Totul despre HTML 4

234215

Rick Darnell

Traducere de Liviu Crăciun, Alexandru Dona,
Ioan Manea și Sorin Bogdan

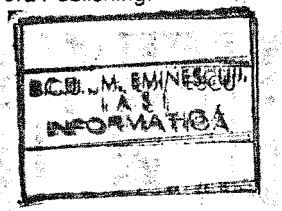
130045
B.C.U. - IASI

Teora

Titlul original: HTML 4 UNLEASHED – Professional Reference Edition

Copyright © 2001, 2000 Teora
Toate drepturile asupra versiunii în limba română aparțin Editurii **Teora**.
Reproducerea integrală sau parțială a textului sau a ilustrațiilor din această carte este posibilă numai cu acordul prealabil scris al Editurii **Teora**.

Authorized translation from the English language edition published by SAMS Publishing
Copyright © 1998
All rights reserved. No part of this book may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying, recording or by any information storage retrieval system, without permission from the Publisher.
Romanian language edition published by Teora Publishing.
Copyright © 1999



Teora
Calea Moșilor nr. 211, sector 2, București
Tel/Fax: 01/210.38.28
e-mail: teora@teora.kappa.ro

Teora – Cartea prin poștă
CP 79 – 30, cod 72450 București, România
Tel: 01/252.14. 31
e-mail: cpp@teora.kappa.ro

Copertă: Gheorghe Popescu
Tehnoredactare: Techno Media
Director Editorial: Diana Rotaru

Președinte: Teodor RĂDUCANU

NOT 4435 CAL HTML 4, TOTUL DESPRE
ISBN: 973-20-0427-4

Printed in Romania

Cuprins

Introducere 9

Partea I **Introducere în HTML 4.0**
1 Scurtă prezentare a versiunii HTML 4.0 14
2 Ce este limbajul de marcare a hipertextului? 19
3 Structura unui document HTML 29
4 Definiția tipului de document HTML 35
5 Acum, să începem 48

Partea a II-a **Elemente de structură și navigație ale HTML-ului 4.0**
6 Structurarea textului 67
7 Alinierea și formatarea textului 81
8 Folosirea listelor pentru organizarea informațiilor 99
9 Organizarea datelor cu ajutorul tabelelor 115
10 Macheta și prezentarea tabelelor 132
11 Adăugarea imaginilor pe pagina dumneavoastră Web 147

Partea a III-a **Interactivitatea de bază a HTML-ului 4.0**
12 Să înțelegem locatorii uniformi de resurse 156
13 Hiperlegături și ancore 167
14 Crearea meniurilor cu hărți imagine 176
15 Crearea și utilizarea formularelor HTML 194
16 Obiecte înglobate în HTML 213

Partea a IV-a **Tehnici de prezentare**
17 Stil simplu cu text 222
18 Utilizarea foilor de stil în cascadă 230
19 Foile de stil în cascadă și poziționarea elementelor 261
20 Crearea cadrelor 273
21 Folosirea foilor de stil JavaScript 287
22 Layere 299

Partea a V-a **Limbaje de programare pentru HTML**
23 Scurtă privire asupra limbajului JavaScript 316
24 Folosirea limbajului VBScript 332
25 Să înțelegem interfața CGI 352

Partea a VI-a **HTML dinamic**
26 În căutarea unui standard 379
27 Modelul de obiect document 390
28 Lucrul cu utilizatorul – Evenimente 399

29	HTML și scripturi – Combinația DHTML	414
30	Asocierea datelor	435
Partea a VII-a	**Designul efectiv al unei pagini Web**	
31	Realizarea unei interfețe eficiente	459
32	Navigarea și machetarea	479
33	Folosirea culorilor și a graficii în WEB	498
34	Folosirea spațiului, a formei și a tipului caracterelor	516
35	Toate puse împreună: forumul DisAbilities din Microsoft Network	539
Partea a VIII-a	**XML**	
36	Ce este limbajul XML?	563
37	Comparație între limbajele XML și HTML	582
38	Utilizarea limbajului XML	590
39	Să înțelegem limbajul XSL	600
Partea a IX-a	**Tehnici avansate**	
40	Internaționalizarea limbajului HTML	623
41	Crearea paginilor Web cu o mare accesibilitate	642
42	Strategii pentru indexare și motoare de căutare	660
Partea a X-a	**Anexe**	
Anexa A	HTML 4.0 – Referințe	670
Anexa B	Tabelul de referințe pentru compararea browserelor	734
Anexa C	Scurtă prezentare a standardului CSS 1	816
Anexa D	Scurtă prezentare a limbajului XML	827
Anexa E	JAVASCRIPT	836
Anexa F	VBScript	870
Anexa G	Resurse pentru dezvoltare	881
Index		945

Despre autori

Rick Darnell provine din câmpiile Kansasului, dar privirea sa este blocată în prezent de o mulțime de munți deoarece locuiește cu soția și cele două fete ale sale în vestul statului Montana. A absolvit Kansas State University cu o diplomă în televiziune, apoi, într-un moment de confuzie, s-a apucat să scrie pentru două reviste tehnice și un ziar săptămânal local. În timp ce își petrecea timpul ca ziarist independent și scriitor, Rick a putut vedea întreaga gamă de calculatoare personale, începând cu Radio Shack Model I, către sfârșitul anilor 1970. Când nu se află în fața calculatorului său de acasă, Rick activează ca pompier voluntar și ca membru al unei echipe responsabile cu manevrarea materialelor periculoase. Rick a scris mai multe cărți pentru Sams.net Publishing și Que Corporation, inclusiv *HTML 4 Unleashed* (cartea de față) și *Teach Yourself Dynamic HTML in a Week*. Adresa sa pe Web este `http://people.montana.com/-darnell/` dar poate fi găsit și prin e-mail la `darnell@montana.com`. În această carte a scris Părțile I, a II-a și a IV-a, precum și Capitolele 16, 26, 27 și Anexa C.

Bruce Campbell (`bdc@hit1.washington.edu`) trăiește în Seattle, WA și se ocupă de tehnologii ale colaborării 3-D bazate pe Web, cum ar fi VRML. Când nu scrie, predă la centrele de instruire Catapult sau la colegiul Central Seattle Community College, realizează cercetări în domeniul VR la Human Interface Technology Laboratory în cadrul Universității din Washington sau aleargă pe undeva prin America de Nord. A colaborat la *Teach Yourself VRLM 2.0 în 21 Days*, *HTML 4 Unleashed* și *Teach Yourself Dynamic HTML in a Week*. Pagina sa URL este `http://www.hit1.washington.edu/people/bdc`. Bruce a scris Capitolele 28, 29 și 30.

David și Rhonda Crowder sunt pionieri în hipertext și fac parte din comunitatea online de mai mult de un deceniu. Compania lor, Far Horizons Software, care se ocupă cu designul paginilor de Web, a creat premiatul LinkFinder și siturile NetWelcome. Ei și pisicile lor trăiesc în Miami, Florida. David și Rhonda au scris Capitolele 12-15 și au contribuit la Anexa G.

Molly E. Holzschlag este instructoare, designer și autoare a numeroase cărți despre designul pe Web și tehnologii legate de Internet, inclusiv *The Laura Lemay Guide to Sizzling Web Site Design* și *Laura Lemay's Web Workshop: Guide to Designing with Style Sheets, Tables and Frames*. Molly predă designul Web studenților care au absolvit facultăți de media la New School for Social Research. Îi puteți viziona situl la `http://www.molly.com/`. Molly a scris Partea a VII-a a acestei cărți.

Will Kelly (`willk@tiac.net`) este un scriitor independent în domeniul tehnic și consultant în zona Washington, DC. El a scris consultația și cursurile pentru proiecte ale unor companii cum ar fi WordPerfect (pe vremea când încă era WordPerfect!), MCI, Oracle, Corporate Software and Technology și Concert. Lucrând adesea ca editor tehnic pentru Sams Publishing, el este mai obișnuit să se plângă față de structura capitolelor decât să scrie un capitol dintr-o carte de informatică. Totuși, pentru această carte, Will a contribuit la Anexa A.

Dmitry Kirsanov a absolvit în 1996 Universitatea din St. Petersburg (Rusia), Facultatea de Litere. Întâi a lucrat ca translator, apoi a urmat o carieră de scriitor independent și a publicat două cărți și numeroase articole pe Internet și Web, toate în limba rusă. De curând a început să scrie pe Web în limba engleză editorialul lunar *Design Lab* la `http://www.webreference.com/dlab/`. Dmitry a scris Partea a IX-a.

Robert McDaniel este un entuziast al tehnologiei Internet și a fost activ pe Web încă din primele momente. Printre lucrările sale se numără *CGI Manual of Style, PC Magazine Webmaster's Ultimate Resource Guide, Late Night Microsoft VisualJ++* și *How to Program Microsoft Visual Basic Scripting Edition*. În prezent lucrează la un proiect intranet la Walt Disney Imagineering Research and Development. Robert a scris Capitolele 24 și 25.

Kelly Murdock (`murdocks@itsnet.com`) lucrează ca programator cu normă plină și își petrece mult din timpul liber explorând frontierele graficii. Îi place în mod deosebit să jongleze cu imagini 3-D. Este autorul principal al cărții *Laura Lemay's Web Workshop: 3D Graphics and VRML 2* și a contribuit la multe alte cărți. El scrie de asemenea un editorial propriu, „Kel's Eye Inside Review" (*Cronica văzută prin ochii lui Kel*), pentru `www.3dreview.com`. Kelly a scris capitolul 23.

Simon North a lucrat în trecut ca tehnician care repara elicoptere, vânzător de asigurări ale calității software-ului și translator tehnic. Este un scriitor tehnic cu experiență, specializat în lucrul online, în multimedia și metode interactive de livrare. Este unul dintre autorii documentației standardului național danez al consumatorului și este coautor al lucrării *Presenting XML*. Simon a lucrat în SGML din 1989 și a fost un participant activ la activitățile de standardizare a CALS/SGML în Danemarca. Simon este un navigator avid pe Internet, construindu-și primul său intranet în 1993. Ultimul său hobby (de când a renunțat la parașutism) este colecționarea conturilor e-mail, dar poate fi contactat la `sintac@xs4all.nl`.

Lee Anne Phillips este una dintre „cei vechi" în afacerile rețelelor, sărind de la Universitatea din California în lumea mainframe din 1977. De atunci a lucrat pentru mai multe companii, jonglând cu diverse aspecte ale lucrului în rețea. Într-un final a ajuns chiar la World Wide Web, în 1994, după un scurt flirt cu ARPAnet în timpul zilelor lipsite de griji ale studenției și o iubire pe termen lung cu Internetul. Acum locuiește în locul său de origine, zona San Francisco Bay din California, după mulți ani de exil sau de călătorii. Deține o licență în informatică și este una dintre acei tocilari legitimați. Interesul său este deocamdată concentrat asupra sistemelor de poștă electronică, interfețelor utilizatorilor și dezvoltării continue a HTML-ului și SGML-ului ca mijloc de comunicare universal. Ea a scris Anexele B și G ale acestei cărți și a avut o contribuție importantă la Anexa A.

BIBL. CENTR. UNIV. „M. EMINESCU" IAȘI
INFORMATICĂ

Introducere

Internetul a progresat extrem de rapid în acești ultimi ani. Odată un tărâm al academicienilor, cercetătorilor și al agențiilor de apărare, Internetul și părțile sale asociate devin foarte repede un mediu utilizat pe larg, un mod de a realiza comunicarea între indivizi, companii și locuitorii din toată lumea.

Ca parte a Internetului, World Wide Web este forța predominantă în dezvoltarea globală a interconectării calculatoarelor. Limbajul său este simplu, interfețele sunt atractive și prietenoase și este adaptabil la o gamă mare de întrebuințări. Există situri Web pentru vânzarea produselor și a ideilor, menținerea aparențelor, informarea publicului, continuarea studiilor și a cunoașterii sau pur și simplu pentru a pierde vremea, iar conceptul World Wide Web este adaptat comunicării interne prin instalarea intraneturilor în interiorul companiilor.

HTML (Limbajul de Marcare a Hipertextului) este limbajul care dă înfățișarea Webului. El constă într-o varietate de elemente numite *etichete* (tags), care pot fi folosite la orice, de la definirea tipurilor de stil și a anteturilor și până la introducerea conținutului specific, cum ar fi imagini, sunete, cuvinte în realitatea virtuală și miniaplicații Java.

Unul dintre marile neajunsuri în lucrul cu HTML este faptul că multe aplicații de editare a paginilor pe Web nu afișează o pagină așa cum ar arăta văzută cu un browser. Ca urmare, autorii și designerii trebuie să se lupte cu un conținut al paginii amestecat cu etichetele care controlează modul în care acesta apare utilizatorului. Vestea bună este că această situație se schimbă rapid, o dată cu dezvoltarea noilor editoare WYSIWYG, care afișează o pagină folosind standardul acceptat de majoritatea browserelor.

Așadar, dacă s-au introdus editoarele grafice, de ce să învățăm complicatul HTML? Există mai multe motive. În primul rând HTML este și un mijloc de organizare, nu numai de design. Etichetele dau structurare și scop fiecărei părți a paginii și explică modul în care aceasta se leagă cu restul documentului. Învățarea organizării care stă în spatele fiecărei pagini conduce la un design mai bun pentru cititori.

În al doilea rând, nici măcar cele mai bune editoare WYSIWYG nu au suport pentru toate etichetele care fac parte din HTML la un moment dat. Uneori este necesar să modifici direct sursa paginii pentru a schimba etichete sau atribute. Pentru a face aceasta, este necesar să cunoști modul în care etichetele se leagă între ele.

Ultimul motiv pentru a învăța HTML este pur și simplu distracția. Capeți o satisfacție sigură când construiești o pagină de Web pornind de la zero. Veți ști de fiecare cărămidă sau scândură care face parte din ea și veți poseda cunoașterea necesară pentru a putea să modificați orice componentă astfel încât rezultatul să fie exact cel pe care l-ați dorit. Pe de altă parte este distracția de a vedea alte pagini de Web și de a ști cum au reușit autorul și designerul să obțină efectul dorit.

HTML CA SUBIECT DE CONVERSAȚIE

World Wide Web devine un subiect de conversație tot mai întâlnit la cină, la petreceri și banchete, în parcare și pe coridoare la locul de muncă. După ce veți citi această carte, suntem siguri că o să descoperiți HTML-ul ca fiind unul dintre cele mai interesante lucruri de pe planetă și veți dori să discutați nuanțele fiecărei etichete cu cei din jurul dumneavoastră.

Să nu faceți acest lucru. Pentru o conversație obișnuită este mai mult decât suficient doar să știți ce înseamnă HTML. Dacă încercați să discutați detaliat despre HTML la o petrecere, lumea se va holba la dumneavoastră, iar în curând vă veți trezi singur într-un colț, cu o farfurie de legume acoperite de un sos prea moale în mână, alături de câinele gazdei care soarbe fiecare cuvânt pe care îl spuneți.

Cum este structurată această carte

Această carte este împărțită în nouă secțiuni diferite, plus șapte anexe. Cartea începe cu fundamentele și conceptele de bază ale HTML-ului pentru ca apoi să se scufunde în detalii despre etichetele standard și atributele acestora, caracteristici avansate și extensii. După ce sunt acoperite detaliile despre limbaj, sunt discutate subiectele adiționale, inclusiv designul, tehnologiile ce stau la bază și noile forme de limbaje de marcare, cum ar fi Extensible Markup Language (limbaj de marcare extensibil).

Iată ce conține fiecare secțiune:

- **Partea I, „Introducere în HTML 4.0"**: Capitolele din această parte acoperă crearea și începuturile HTML-ului, începând de la conceptul de hipertext și extinzându-se până la tipuri de documente, structuri de documente și cum se potrivesc pe Web diferite browsere și platforme. Capitolul 5, „Acum, să începem", a fost adăugat pentru a vă iuți ritmul dacă aveți deja cunoștințe despre HTML 3.2.
- **Partea a II-a, „Elemente de structură și navigație ale HTML-ului 4.0"**: Această secțiune acoperă etichetele standard suportate de ultimele specificații HTML 4.0; include elemente intens folosite, cum ar fi etichete divizionare și organizaționale, liste și tabele. De asemenea include elemente care ar trebui să fie folosite, dar nu sunt, ca de exemplu declarațiile tipului de document.
- **Partea a III-a, „Interactivitatea de bază a HTML-ului 4.0"**: Pe o pagină de Web se pot face mai multe lucruri decât să citești niște cuvinte. Această secțiune discută despre fundamentele construirii interactivității pe o pagină, folosind elemente cum ar fi ancore, hărți-imagini, formulare și elemente cum ar fi miniaplicații sau controale înglobate.
- **Partea a IV-a, „Tehnici de prezentare"**: În continuarea celor două secțiuni anterioare, care acoperă structura și rolul unei pagini Web, această parte prezintă ultimele tehnici pentru ca pagina să arate exact așa cum vă doriți. Începe cu opțiunile de formatare bazate pe etichete, cum ar fi <B> pentru bold și <I> pentru italic, pentru ca apoi să progreseze către foi de stil în cascadă, foi de stil JavaScript, poziționarea elementelor, cadre și elemente de așezare în pagină.
- **Partea a V-a, „Limbaje de programare pentru HTML"**: Capacitățile HTML-ului s-au extins mult datorită suportului oferit de tehnologii care adaugă caracteristici interactive unor pagini altfel fără viață. Aceste tehnologii includ Comon Gateway Interface (CGI), scripturi pentru transfer de date JavaScript și VBScript (Visual Basic Script).
- **Partea a VI-a, „HTML dinamic"**: Dynamic HTML este o nouă metodă de a integra HTML, scriptarea și foile de stil pentru a crea pagini de Web interactive fără ca acestea să fie procesate prin serverul Web, miniaplicații sau alte soluții adăugate. Cel mai bun lucru este că, fiind corect implementat, Dynamic HTML poate fi văzut cu orice browser, cu caracteristicile dinamice activate doar pe browserele compatibile.
- **Partea a VII-a, „Designul efectiv al unei pagini Web"**: Cu pregătirea deja efectuată din secțiunile anterioare, aveți cunoștințele tehnice pentru a construi pagini Web cu ajutorul HTML-ului. Totuși, o pagină corectă din punct de vedere tehnic nu este neapărat o pagină prietenoasă cu utilizatorii. Această secțiune vă arată cum să construiți pagini atractive și folositoare prin includerea unor concepte și exemple pentru a realiza interfețe de utilizator și metode de navigare.
- **Partea a VIII-a, „XML"**: Limbajul de marcare extensibil este noul puști în cartierul Internetului, iar folosirea sa se extinde datorită inerentei flexibilități. Rezultatul este că XML vă permite să creați pagini cu propriile dumneavoastră etichete personalizate de marcare. Această caracteristică este deja folosită într-una din ultimele realizări pe Web – difuzarea și subscrierea la canale pe Web. Această secțiune explică ce este XML și cum poate fi implementat pe situl dumneavoastră.
- **Partea a IX-a, „Tehnici avansate"**: Capitolele din această secțiune discută evoluția standardelor pentru seturile de caractere și reprezentarea paginii pe plan internațional. De asemenea, sunt tratate problematicile standardizării browserelor și platformelor, precum și folosirea HTML-ului în medii non-Web, cum ar fi motoarele de indexare și de căutare. În plus, conține un capitol întreg dedicat problemelor de accesibilitate – pentru browsere și persoane mai mult sau mai puțin capabile.

Anexele de la sfârșitul cărții includ mai multe referiri informative pentru a vă ajuta în munca de zi cu zi folosind HTML-ul. În plus față de o scurtă referire la etichetele HTML, anexele includ referiri la HTML 4.0, foile de stil în cascadă (nivelul 1), XML și limbajele de scriptare. Sunt furnizate informații adiționale în vederea comparării browserelor, culori și valori ale caracterelor și resurse de informații online.

Sperăm ca această combinație de informații din capitole și referiri în anexe să facă din lucrarea de față cea mai completă carte de referințe despre HTML pe care o aveți în bibliotecă.

Cum să folosiți această carte

Fie că sunteți veteran sau începător, Capitolele 1-5 reprezintă o introducere bună și în același timp o explicare a modului în care arată și acționează HTML-ul și de ce, precum și a funcționării acestuia în lucrul cu cadre pe Internet și pe intraneturi.

Capitolele de la 6 la 16 sunt esența HTML-ului. În această parte este explicată fiecare etichetă HTML, precum și atributele lor și modul în care se comportă. Dacă sunteți începător, parcurgând aceste capitole veți căpăta cunoștințe solide pentru lucrul cu HTML. Aceste capitole sunt susținute de Capitolele 17-22, care vă furnizează un număr mare de metode de a controla înfățișarea paginii Web.

Capitolele 31-35 oferă informații suplimentare despre tehnicile de prezentare. Acest curs de design pe Web conține exemple care vă vor face în cel mai scurt timp să construiți pagini Web cu o structură bună. Dacă ați mai lucrat în HTML, ar trebui să treceți în revistă această parte pentru a vedea noi metode de a prezenta informațiile utilizatorilor într-un mod care să arate bine și să le atragă atenția.

Pentru utilizatorii experimentați, Capitolele 23-30 și 36-39 arată cum să extindă HTML-ul dincolo de elementele ilustrate în capitolele precedente. Aici sunt prezentate și explicate concepte avansate, inclusiv Dynamic HTML și XML.

Ultima parte, Capitolele 40-42, este indicat să o parcurgeți, indiferent de experiența dumneavoastră în HTML. Aici sunt prezentate metode de a face paginile dumneavoastră HTML să fie mai prietenoase pentru publicul internațional. Vă ajută, de asemenea, să creați pagini care să nu poată fi accesate doar de „cei doi mari", Netscape și Microsoft, ci și de utilizatorii defavorizați de browserele lor cu scop specificat. Ca un ultim pas, sunt discutate și metode de indexare a paginilor și adăugarea lor motoarelor de căutare.

ATRIBUTE COMUNE

Majoritatea elementelor din această carte au în comun anumite atribute. În loc să le prezentăm pentru fiecare element în parte, le-am scris pe toate aici, iar în capitolele următoare voi face doar referiri la ele. Primul set de atribute este cel al atributelor principale:

`id`: Această valoare servește ca identificator în cadrul documentului; el trebuie să fie unic, deci diferit de orice alt ID în cadrul documentului. *(continuare)*

(continuare)

`title`: Acest atribut furnizează informații adiționale despre element și poate fi folosit sau ignorat, după cum dorește browserul.

`class`: Foile de stil folosesc acest atribut pentru a aplica un stil unui set de elemente asociate cu această clasă. Vezi Capitolul 18, „Utilizarea foilor de stil în cascadă", pentru mai multe informații.

`style`: Acest atribut este folosit pentru a introduce informații inline despre stilul aparției actuale a elementului. Pentru mai multe informații vezi Capitolul 18.

Următoarele două sunt atribute internaționale (vezi Capitolul 40, „Internaționalizarea limbajului HTML"):

`lang`: Acest atribut specifică limbajul primar al textului din document, având valoarea prestabilită unknown (necunoscut). Pentru mai multe informații consultați „Tags for the Identification of Languages" („Etichete pentru identificarea limbajului"), disponibil la `ftp://ds.internic.net/rfc/rfc1766txt`.

`dir`: Direcția unui text slab sau neutru, care poartă valoarea `LTR` (de la stânga la dreapta) sau `RTL` (de la dreapta la stânga). Browserul determină direcția prestabilită, însă pentru engleză și limbile europene putem să presupunem cu siguranță că este `LTR`.

Următoarele sunt atributele de aliniere, care determină modul în care textul este așezat în cadrul documentului. Valorile sunt `left`, `right`, `center` sau `justify`. În mod prestabilit sunt prevă- zute `left` pentru paragrafe `LTR` și `right` pentru paragrafe `RTL`. Multe browsere nu recunosc `justify` (atât stânga, cât și dreapta), caz în care alinierea prestabilită are valoarea dată de atributul **`dir`**.

Ultimele atribute sunt cele care manevrează evenimentele – `onclick`, `ondblclick`, `onmousedown`, `onmouseup`, `onmouseover`, `onmousemove`, `onmouseout`, `onkeypress`, `onkeydown` și `onkeyup` – care sunt discutate mai detaliat în Capitolul 28, „Lucrul cu utilizatorul – evenimente".

Convenții folosite în această carte

Sams.net a publicat de-a lungul multor ani cărți de calculatoare special concepute pentru a fi ușor de întrebuințat și care conțineau informații de ultimă oră. Pe baza acestei experiențe au fost stabilite caracteristicile care ajută cel mai mult cititorul. Căutați aceste caracteristici prin tot cuprinsul cărții, pentru a vă ajuta să vă dezvoltați lucrurile învățate și să profitați la maxim de HTML.

- Mesajele pe ecran, listarea codurilor și exemplele de comenzi apar cu caractere `monospace`.
- Adresele URL (Uniform Resource Locators) folosite pentru identificarea paginilor pe Web și valorile atributelor sunt scrise de asemenea cu caractere `monospace`.
- Termenii definiți în text apar cu caractere *italice*. Caracterele italice sunt câteodată folosite și pentru a scoate ceva în evidență.
- În liniile de cod, elementele care țin locul variabilelor sunt indicate prin folosirea tipului *`italic monospace`*.

SUGESTIE

Sugestiile vă dau sfaturi asupra procedurilor mai rapide sau trecute cu vederea, inclusiv scurtături.

OBSERVAȚIE

Observațiile prezintă informații folositoare sau măcar interesante care nu sunt esențiale în subiectul respectiv, dar vă pot ajuta să înțelegeți oferindu-vă sfaturi sau cunoștințe legate de subiectul în cauză.

ATENȚIE!

Aceste paragrafe vă previn despre ce probleme potențiale ar putea cauza procedura respectivă, rezultate neașteptate sau greșeli care pot deveni grave.

PERICOL!

Aceste paragrafe sunt asemănătoare cu cele de tip ATENȚIE! doar că sunt ceva mai grave. Ele vă vor avertiza despre acțiuni care pot deranja sau deteriora sistemul cuiva.

Cine ar trebui să citească această carte?

Această carte a fost special concepută și realizată pentru a corespunde unei game mari de necesități, în funcție de nivelul dumneavoastră de experiență și de pregătire în HTML.

Pentru începători, prezentăm o introducere în bazele HTML-ului, inclusiv structura de bază a unei pagini și toate etichetele necesare pentru a construi una. Fiecare element HTML este prezentat însoțit de atributele corespunzătoare, comportamentul standard și minimul de informații de care are nevoie pentru a funcționa.

Atât utilizatorii instruiți, precum și cei ocazionali o să considere mai ușor să sară direct la un subiect anume de care au nevoie, cum ar fi tabele, cadre sau design. Amintiți-vă că această carte este o lucrare completă despre HTML 4.0, astfel încât ar fi bine să aruncați o privire și la celelalte capitole, ca să vedeți ce ați scăpat din vedere.

Pentru experți, această carte poate să reprezinte o excelentă sursă de referințe pentru a le răspunde la întrebările specifice. Sintaxele, listele de atribute și exemplele o să vă furnizeze multe oportunități de a vedea variațiuni pe tema implementării HTML-ului. Referințele de la sfârșitul cărții plasează toate informațiile tehnice necesare la îndemâna oricui.

Am muncit mult să asamblăm cea mai completă carte de HTML disponibilă și sperăm că veți considera această carte nu numai o adăugare importantă în bibliotecă, ci și o unealtă folositoare în lucrul de zi cu zi.

Din partea tuturor autorilor vă mulțumim că ați ales această carte.

— Rick Darnell

(`darnell@montana.com`)

PARTEA I

Introducere în HTML 4.0

Capitolul 1
Scurtă prezentare a versiunii HTML 4.0

de Rick Darnell

În acest capitol:

- Noi caracteristici 15
- Caracteristici nerecomandate 17
- Caracteristici moarte 17

Bine ați venit în minunata lume nouă a ultimelor recomandări făcute de consorțiul World Wide Web privind limbajul HTML. Deși aceste specificații sunt mai greu de înțeles pentru cei care folosesc Webul ocazional, ele sunt cu adevărat interesante.

HTML 4.0 reia părțile eliminate din HTML 3.2. HTML 3.2 și-a început existența ca HTML 3.0 și conținea propuneri pentru multe dintre caracteristicile găsite în HTML 4.0. Totuși, momentul nu a fost bine ales pentru HTML 3.0 – a fost respins în procesul de aprobare și, oficial, nu a fost niciodată acceptat ca standard. Nu s-a căzut de acord asupra modului în care noile caracteristici aveau să fie reprezentate în standard și cum vor fi acestea interpretate de diverse browsere. Așadar a fost eleborat un nou standard, HTML 3.2, care lăsa la o parte majoritatea caracteristicilor propuse inițial pentru HTML 3.0, inclusiv setările avansate ale tabelelor, evenimentele și altele. În loc să conducă la evoluția HTML-ului, procesul de standardizare a devenit o reflectare a practicilor acceptate și adoptarea mărcării de proprietar.

Acum, membrii W3C HTML Working Group și ai altor grupuri și subcomisii au avut șansa de se regrupa și de a-și concentra eforturile. Rezultatul a fost o nouă specificație HTML, care se întoarce la rădăcina unui limbaj de marcare structurat, în același timp implementând puternice mijloace de control al aspectului și capacității pentru comportarea dinamică viitoare.

OBSERVAȚIE

Cartea acoperă ultimele informații disponibile privind versiunea preliminară a specificației HTML 4.0, cunoscută ca *WD-html40 – 970917*. Aceasta este abreviarea pentru „Working draft of HTML 4.0. as of September 9, 1997". Este o versiune preliminară care, în linii mari, va rămâne neschimbată înainte de a fi acceptată ca o recomandare oficială. Pentru mai multe informații despre statutul său, vizitați situl Web – W3C (vezi următoarea sugestie).

SUGESTIE

Situl World Wide Web Consortium are adresa `http://www.w3.org/`. Situl include recomandări, ultimele propuneri și informații despre proiectele W3C, cum ar fi browserul Amaya Web.

Dar tot ce a fost până acum sunt doar chestiuni generale. Citiți și paginile următoare pentru a vedea ce s-a întâmplat de la apariția standardului HTML 3.2.

Noi caracteristici

Sunt câteva noi etichete și caracteristici în HTML 4.0. Una dintre primele diferențe notabile este suportul pentru identificare și capturarea evenimentelor pentru aproape fiecare element din corpul unei pagini de Web. Atributul `id` este suportat pentru tot, de la titluri la tabele, elemente de formular sau citate de bloc. După ce atribuiți un nume unic unui element, puteți să faceți referiri către acesta prin ancore și caracteristicile Dynamic HTML.

Acoperirea extinsă a elementelor a fost introdusă de Netscape în versiunea Navigator 2.0 cu JavaScript, numai că evenimentele erau limitate la câteva etichete – în principal hiperlegături și elemente de formular. Începând cu Dynamic HTML și scriptarea întregului document, evenimentele sunt acum suportate pentru aproape orice element, indiferent de conținutul său, cum ar fi imagini, ancore, paragrafe, tabele (inclusiv rânduri individuale, celule și coloane) și alte elemente inline sau de bloc.

Altă apariție nouă în HTML 4.0 este trecerea de la caracteristicile privind aspectul bazat pe etichete (cum ar fi `<B>` pentru bold) la foile de stiluri. Această trecere este rezultatul unei lungi dezbateri între puriștii HTML-ului și designerii de pagini Web. Primii insistă ca etichetele să indice doar structurile și conținutul – datorită varietății incredibile de aplicații și platforme pentru vizualizarea paginilor, autorul nu ar trebui să fie în stare să hotărească modul în care va fi prezentată o pagină. De cealaltă parte se află designerii, care profită de fiecare ciudățenie și particularitate a implementărilor HTML pentru a face ca paginile lor să arate exact cum trebuie; concurența este mare și vor să se asigure că paginile lor sunt în frunte.

Neapreciind etichetele și atributele care afectează aspectul (vezi secțiunea următoare, „Caracteristici nerecomandate"), puriștii se întorc la un HTML mai bun și mai blând, care nu permite folosirea simultană a etichetelor `<EM>` și `<I>` pentru o evidențiere într-un document. Prin adăugarea foilor de stil, designerii au acum la dispoziție un nou set de ustensile pentru crearea paginilor lor, incluzând specificații avansate pentru fonturi, culori de fundal pentru elemente individuale și stiluri globale. Un avantaj al acestei dezvoltări este că cele mai bune pagini pot fi vizualizate și cu Lynx, și cu Internet Explorer 4.0, iar utilizatorii acestor browsere vor beneficia din plin de conținut fără ca o mulțime de etichete să le stea în drum.

Eticheta `<OBJECT>` a fost de asemenea extinsă, astfel încât să includă practic orice fișier extern pe care l-ați putea dori dintr-o pagină Web, inclusiv miniaplicații, controale înglobate, plug-ins-uri, imagini, sunete și tot ce vă mai trece prin minte. Cu lista sa extinsă de atribute și cu suportul pentru parametri și text alternativ, eticheta `<OBJECT>` înlocuiește alte etichete, cum ar fi `<EMBED>`, `<APPLET>` și `<IMG>`. O facțiune încearcă să folosească această etichetă chiar și în loc de `<IFRAME>` pentru cadre inline.

În afară de implicațiile globale ale noilor atribute și foi de stil, HTML 4.0 introduce mai multe etichete de structură, prezentate în Tabelul 1.1.

Tabelul 1.1. Noi etichete de structură în HTML 4.0.

Eticheta	*Descriere*
`<Q>`	Pentru citate inline. `<Q>` este o rudă a etichetei `<BLOCKQUOTE>`, `<Q>` fiind un element inline, iar `<BLOCKQUOTE>` comportându-se ca un paragraf în sine.
`<INS>` și `<DEL>`	Pentru documente la care încă se lucrează. Sunt importante pentru istoria modificărilor. `<INS>` atribuie data inserării textului adăugat, iar `<DEL>` face același lucru pentru textul șters.
`<ACRONYM>`	O nouă etichetă de structură, similară cu `<ADDRESS>`. Marchează un grup de litere care reprezintă altceva, cum ar fi *SCBA* pentru *self-contained breathing aparatus*.
`<COLGROUP>`	Identifică o colecție de coloane într-un tabel pentru formatare. Permite designerilor să selecteze ușor coloanele într-un tabel pentru a seta proprietățile legate de lățime și aliniere.
`<FIELDSET>`	Grupează elemente de formular în grupuri logice. Această etichetă este însoțită de eticheta `<LEGEND>`, care furnizează un ecuson vizibil unui grup de controale de formular.
`<BUTTON>`	Un buton bogat în acțiuni în ceea ce privește formularele. Acesta este bun la toate, dincolo de butoanele Submit și Reset. El poate să preia controlul asupra formularelor, să calculeze totalul, să invoce scripturi, să interacționeze cu obiecte și include formatări variate, cum ar fi culori, diferite fonturi și mărimi ale textului, precum și imagini.

Ultimul lucru adăugat, și cel mai controversat, este reprezentat de cadre. HTML 4.0 va include seturile de cadre și cadrele ca parte din standardul oficial, spre disperarea multor utilizatori de Web și autori care nu le pot suferi. Totuși W3C a hotărât să le folosească deoarece erau suficient de utilizate pentru a standardiza ce exista deja, astfel încât să se poată lucra la o mai bună implementare a ideii de cadru. Fie că iubiți cadrele, fie că le detestați, acum ele sunt în mod oficial parte din HTML.

Bineînțeles, cu toate înfloriturile adăugate noului HTML, acesta conține niște etichete și atribute care nu se potrivesc destul de bine. Acestea sunt elementele *nerecomandate* – există în standard, dar nu pentru mult timp.

Caracteristici nerecomandate

Ce este o *caracteristică nerecomandată*? Este o etichetă sau un atribut care a făcut parte dintr-un HTML mai vechi, dar de care nu mai este nevoie în continuare sau existența sa este în detrimentul progresului HTML-ului. Acestea rămân încă în standard la ediția actuală, dar probabil vor fi lăsate la o parte sau declarate depășite la următoarea ediție a HTML-ului.

SUGESTIE

Dacă folosiți caracteristici nerecomandate și procesați aceste documente prin validarea HTML de pe situl W3C, datorită strictețiii HTML 4 Document Type Definition (vezi Capitolul 4, „Definiția tipului de document HTML"), pagina va genera erori. Elementele depreciate fac parte doar din versiunea „obișnuită" a definiției tipului de document HTML.

Vă amintiți toată tevatura iscată de foile de stil? Înfățișarea și structura unui document sunt în sfârșit separate și plasate în locații diferite. Însă această separare înseamnă de asemenea că etichete cum ar fi `<FONT>` nu mai au nici o întrebuințare, datorită faptului că acțiunile lor sunt acum acoperite de foaia de stil. Printre etichetele de aspect care acum s-au depreciat se numără `<CENTER>`, `<FONT>`, `<BASEFONT>`, `<STRIKE>` (la fel `<S>`) și `<U>`. Este un fapt destul de sigur că și alte etichete de aspect, cum ar fi `<B>` și `<I>`, se îndreaptă către moarte sigură în viitoarea versiune de HTML.

Între timp vedeți Capitolul 17, „Stil simplu cu text", pentru a afla cum să lucrați cu ce a mai rămas, și Capitolul 7, „Alinierea și formatarea textului", pentru informații despre cum să lucrați cu stiluri logice.

Pe lângă etichetele de aspect nerecomandate, atributele pentru aspect ale multor elemente sunt și ele pe cale de dispariție. Acestea conțin informații despre culoare care pot fi incluse și cu eticheta `<BODY>`, cum ar fi `bgcolor`, `color` și `link`. Atributul `align` pentru paragrafe și alte etichete de blocuri s-a depreciat din aceleași motive.

Alte elemente deja selectate pentru a fi excluse sunt următoarele:

- **`<APPLET>`**: Introdus de Netscape pentru a susține miniaplicațiile Java, rolul său fiind acum acoperit de eticheta `<OBJECT>`, care s-a extins pentru a servi tuturor sarcinilor de includere a unui fișier (vezi Capitolul 16, „Obiecte înglobate în HTML")
- **`<ISINDEX>`**: Această etichetă este folosită pentru a genera un câmp de text care să introducă un termen de căutare într-un index de Web. Această funcție a fost acum atribuită formelor și scripturilor CGI (vezi Capitolul 15, „Construirea și folosirea formularelor HTML").
- **`<DIR>`, `<MENU>`**: La bază, aceste elemente copiază funcțiile etichetei de listă neordonată, `<UL>` (vezi Capitolul 8, „Folosirea listelor pentru organizarea informațiilor"). Au fost plasate în această categorie deoarece majoritatea browserelor nu făceau diferența între aceste două etichete și o listă neordonată.

Caracteristici moarte

Acesta este cimitirul elementelor depășite, care nu se mai regăsesc nicăieri în standardul HTML. Într-un an sau doi o să fiți norocos să găsiți un browser care încă să le mai suporte.

Cele trei elemente care au fost excluse din HTML 4.0 sunt `<XMP>`, `<PLAINTEXT>` și `<LISTING>`. Toate trei au același rezultat: text preformatat care prezintă caracterele, punctuația și spațiile cu un font monospațiat. Exact aceeași funcție o are și elementul `<PRE>`, care încă este o parte activă în HTML. Toate cele trei elemente erau deja nerecomandate în HTML 3.2, iar scoaterea lor oficială din standard s-a făcut în HTML 4.0.

Rezumat

Ați citit deja despre schimbările făcute. Este HTML 4.0 varianta perfectă? Desigur că nu. Câtă vreme este folosit de mai mulți oameni vor exista păreri diferite despre ce ar trebui sau nu să facă parte din standard.

Consensul general în comunitatea Web este că HTML 4.0 a făcut un pas înainte față de HTML 3.2 cu implementarea foilor de stil pentru controlul aspectului și că este un model extins pentru tabele, suport pentru noile tehnologii Dynamic HTML și îmbunătățirea formularelor.

Datorită faptului că elementele pentru aspect se vor deprecia în viitor, HTML va continua să se axeze pe marcarea structurii, iar controalele pentru aspect se vor dezvolta în limitele impuse de eticheta <STYLE>.

Una peste alta, este un standard solid, care a fost foarte bine evaluat și implementat. Are defctele sale, dar ce a fost mai rău în evoluția HTML-ului s-a terminat. Ca rezultat, autorii și creatorii de pagini de Web s-au ales cu un nou și mai eficient grup de etichete pentru crearea paginilor de Web.

Mă bucur că am putut să vă spun aceste lucruri! Mai sunt numai 41 de capitole, așa că hai să începem...

Capitolul 2
Ce este limbajul de marcare a hipertextului?

de Rick Darnell

În acest capitol:

- **Lucrul cu informația neliniară 19**
- **Încercări timpurii de utilizare a sistemelor de informație neliniară 20**
- **Originile HTML-ului 21**
- **World Wide Web – primul sistem practic de informație neliniară 22**
- **Crearea HTML-ului 22**
- **Consorțiul World Wide Web 27**

Deși mulți cred că HTML-ul constă în etichete puse într-o pagină de World Wide Web, conceptele care au stat la baza HTML-ului au o istorie lungă. Acest capitol prezintă pe scurt unii oameni și idei care au contribuit la apariția Web-ului nostru, cu text, imagini, animație, sunet și alte posibilități.

Voi începe cu un concept – informația neliniară – pentru a vă oferi un punct de plecare în învățarea HTML-ului, care este utilizat la realizarea paginilor actuale de Web. Luați-vă inima în dinți – nu este atât de tehnic și de sec pe cât pare.

Lucrul cu informația neliniară

Ce este informația neliniară? Pentru a vă răspunde să ne întoarcem puțin și să ne gândim la informația liniară. Aceasta este una dintre cele mai uzuale modalități prin care comunicăm între noi. „Liniar" este un adjectiv generat de cuvântul „linie", care sugerează legătura sau asemănarea cu o linie dreaptă. Așadar, informația liniară trebuie să fie informația care progresează în linie dreaptă, începând într-un punct A și terminându-se într-un punct B, urmând un traseu stabilit între acestea.

Un roman este un exemplu de informație liniară. Începeți cu prima pagină și continuați în ordine către sfârșit. Acest mod de a citi prezintă informația într-o linie, percepând desfășurarea acțiunii și personajele într-o ordine prestabilită, așa că la sfârșit totul are un sens.

Ați putea citi un roman într-o manieră neliniară? Desigur. Trebuie doar să alegeți pagini la întâmplare până când ați citit întreaga carte. Romanul a fost citit, dar, datorită faptului că informația nu v-a parvenit în ordinea menită, există multe șanse să nu aveți o concepție reală asupra conținutului cărții.

Înseamnă aceasta oare că informația neliniară creează confuzie? Nu neapărat. Totul depinde de modul în care această informație este folosită. O revistă este un exemplu bun pentru o sursă de informații care profită de structura neliniară. Folosind cuprinsul, puteți deschide revista la pagina care vă interesează. Probabil că veți întâlni expresii cum ar fi „continuarea la pag. 58", care vă vor îndruma către urmarea articolului aflată peste câteva pagini (după niște reclame importante) sau chiar aproape de sfârșitul revistei. Acest mod de a citi este neliniar, deoarece nu trebuie să citiți tot ce se află între cuprins și începutul

articolului pentru a înțelege conținutul acestuia. La fel, nu trebuie să citiți paginile dintre punctul în care articolul era întrerupt și locul în care continuă.

O analogie similară există într-un format muzical. O casetă cu muzică este un format liniar. Ea este menită să o ascultați de la un capăt la altul. Puteți încerca să săriți direct la melodia favorită, aflată la sfârșitul casetei, dar tot trebuie să treceți peste toate celelate melodii care sunt înaintea ei. În schimb un CD este un format neliniar. Puteți selecta orice melodie de pe CD și o puteți asculta aproape instantaneu, fără să așteptați să treceți prin toate celelalte melodii.

Încercări timpurii de utilizare a sistemelor de informație neliniară

Au fost puse pe hârtie întregi teorii și concepte despre informația neliniară, dar hârtia în sine a fost neajunsul tuturor sistemelor propuse. Cerința ca sistemele de informație neliniară să fie dependente de hârtie a condamnat marea lor majoritate la un inevitabil eșec. Câteva au rezistat, și rezistă încă, cum ar fi cutiile cu fișe de carton sau dulapurile cu dosare, dar majoritatea au dispărut.

Oamenii au încercat diverse soluții pentru a face aceste sisteme neliniare chiar și mai accesibile, cum ar fi inventarea cartelelor indexate, pentru a ajuta la organizarea cercetării. O companie a făcut cartele cu găuri de-a lungul muchiilor, permițând utilizatorilor să le transforme în crestături. Găurile erau numerotate, astfel încât utilizatorii le puteau atribui anumitor subiecte. O persoană putea să introducă un ac printr-un pachet de cartele în gaura respectivă și, scuturând pachetul, făcea astfel să cadă cartelele cu crestături și să rămână pe ac doar cele selectate. Acele cartele puteau fi apoi sortate din nou, selectându-le și scuturându-le în runde consecutive, pentru ca, în cele din urmă, să rămână doar cele necesare lucrării respective.

Bineînțeles, era nevoie de multă muncă pentru a pune în practică acest sistem. Cercetătorii și asistenții trebuiau să compileze informațiile pe cartele, să creeze indexul, să taie cartelele exact unde trebuia și așa mai departe pentru a avea un sistem folosibil. Dacă vroiau să folosească sistemul în comun, atunci fiecare trebuia să aibă aceleași metode de tăiere și indexare, iar fiecare cartelă scoasă de unul din ei devenea indisponibilă pentru ceilalți.

Enter Vannevar Bush a fost directorul Biroului de Cercetare Științifică și Dezvoltare sub Franklin Roosevelt în timpul celui de-al doilea război mondial. Bush a construit mașini prevăzute cu aparate optice pentru selectarea rapidă a anumitor role cu microfilm și și-a folosit poziția în timpul administrației Roosevelt pentru a-și rafina ideile sale despre stocarea și recuperarea unor mari cantități de date folosind legături directe către utilizator. A folosit termeni precum *trails* (trasee) sau *footprints* (amprente) pentru a descrie calea dintre utilizator și diverse surse de informații.

După război, Bush a descris o mașinărie pe care am putea-o considera ca fiind primul browser. Era o mașinărie incredibil de complicată, care consta într-un birou, ecrane translucide de proiecție puse deasupra, o tastatură, butoane și manete. Conținutul – cum ar fi cărți, reviste și hârtii de la cercetători – era achiziționat pe microfilme. Biroul avea de asemenea posibilitatea să înregistreze lucrări, cum ar fi notițe sau diagrame, pe microfilm pentru a fi vizionate mai târziu. Cărțile erau chemate prin introducerea unui cod, apoi proiectate pe ecran în timp ce un mâner controla viteza la care se schimbau paginile. Când era creat un nou material, i se atribuia un cod care era stocat într-un index la un loc cu codurile celorlalte lucrări aflate în conținut.

Tot Bush a făcut următorul pas în ceea ce privește indexarea, permițând indexarea asociată. Utilizatorii puteau determina orice obiect, în orice moment, să selecteze imediat alt obiect. Principiul era că un utilizator putea alcătui un traseu la care erau atașate obiecte ce puteau fi reapelate instantaneu doar printr-o apăsare de buton. Traseul putea să includă parțial sau să interconecteze obiecte, astfel încât, o dată creat, un utilizator putea să-l urmeze după bunul plac, să facă investigații parțiale sau colaterale.

Originile HTML-ului

De la fantasticul birou interactiv facem un salt către anii 1960, la un om pe nume Ted Nelson, care deținea o diplomă în filozofie și se întorsese la școală pentru licența în sociologie. În timpul unui curs despre calculatoare ținut pentru studenții umaniști, și-a dat seama de potențialul acestora în domenii precum gândirea și scrisul. Acestea se întâmplau în era calculatoarelor mainframe, a cartelelor găurite și a cititoarelor de bandă perforată, astfel încât legătura nu era evidentă pe atunci.

Nelson a continuat această idee și a imaginat un sistem pe care l-a denumit Xanadu, care includea pentru prima oară folosirea cuvântului *hipertext*. În această lucrare, cât și în cele care au urmat, el a propus scrierea neconsecutivă, apreciind că scrierea a fost mereu consecutivă deoarece paginile erau consecutive și deci nu exista altă variantă. Nelson a propus utilizarea calculatorului pentru a ne propulsa mințile în hiperspațiul gândirii. El a denumit noua lume de texte și imagini instantaneu disponibile *hyperworld* (hiperlume), iar sistemul de stocare *hyperfile* (hiperfișier).

În ciuda abordării inovative și creatoare, Xanadu nu a fost niciodată realizat. Nelson a anunțat că acesta va fi gata întâi în 1976, apoi în 1988, 1991 și 1995, nici un termen nefiind respectat. Revista *DATAMATION* a scris despre Xanadu ca fiind primul exemplu de vaporware.

Printre problemele pe care le-a întâmpinat Xanadu se numără inventarea unei multitudini de termeni și schimbările de terminologie, care i-au marcat progresul. Nelson a folosit termeni cum ar fi *xanalogics, humbers, docuverses* și *tumbler arithmetic*. Înțelegerea acestor noi termeni și ținerea pasului cu noile denumiri au făcut ca acceptarea largă a acestui proiect să fie o bătălie pierdută din start. Totuși, conceptele sale există astăzi în World Wide Web, pe când sistemul Xanadu propriu-zis pare că a murit, rămânând doar un nume.

În altă arenă se afla Bill Atkinson, care a creat o mulțime de programe superbe pentru calculatoarele Apple Macintosh, inclusiv MacPoint și MultiFinder. În 1987, HyperCard folosește aptitudinile grafice ale calculatorului Macintosh pentru a arăta un pachet de cartele pe ecran. Cartelele în sine pot conține un text, sunet, filme sau imagini, împreună cu butoane și alte mijloace de navigare, pentru ca utilizatorii să poată trece de la o cartelă la alta.

Hipercartelele (HyperCards) sunt create într-un limbaj interpretat, denumit *HyperTalk*, care este similar cu limba engleză și poate fi citit și înțeles fără să studiați prea mult. Folosind HyperTalk, autorii de scripturi pot crea cartele, să accepte intrările de la utilizator prin intermediul tastaturii sau al mouse-ului, să reacționeze la intrări și să afișeze rezultatele. Utilizatorii au scriptat programe și jocuri complexe folosind HyperTalk, deși acesta poate fi destul de lent.

HyperCard prezintă câteva neajunsuri, printre care și lipsa conexiunilor. Singura opțiune pe care un utilizator o are pentru conexiuni e reprezentată de butoane, care trebuie așezate astfel încât să nu acopere textul de pe ecran. Pentru a adăuga text unei cartele poate fi necesar ca autorul să rescrie locațiile tuturor butoanelor de navigare. La început, HyperCard suporta numai o mărime de cartelă, în alb-negru, fiind vizibilă o singură cartelă la un moment

dat. Similar „biroului integrat" al lui Bush de la sfârșitul anilor '40, HyperCard era în continuare restrâns la universul calculatorului pe care era rezident.

In ciuda limitării acestora, cartelele au acoperit o necesitate la vremea aceea și au devenit foarte cunoscute. Conceptul HyperCard a ajuns la maturitate datorită furnizorilor, care l-au dezvoltat. Este vândut în prezent ca un produs separat al firmei Macintosh.

Următorul pas important a fost corelarea funcțiilor HyperCard-ului cu conceptele introduse de Xanadu, fapt care s-a petrecut în final în Laboratorul European de Fizică a Particulelor (CERN).

World Wide Web – primul sistem practic de informație neliniară

Cu multe dintre ideile, echipamentele și programele necesare la un loc pentru prima oară, sistemul de informație neliniară era în sfârșit posibil. Soluția a venit de la Tim Berners-Lee, un om de știință englez care lucra la CERN. El a observat dificultățile pe care le întâmpinau oamenii de știință din diferite țări în comunicarea dintre ei. Deși fusese angajat să lucreze la un accelerator de particule de înaltă frecvență, s-a implicat în facilitarea comunicării dintre oameni.

Mai devreme în cariera sa, acesta a scris un program numit Enquire, care permitea să facă asociații în notițele sale personale. Acest program a fost realizat pentru ca luarea notițelor și păstrarea acestora să fie realizată într-un mod mai apropiat de cel în care gândim. Această idee s-a transformat într-un nou concept, de a folosi calculatoarele pentru a-i ajuta pe utilizatori să caute într-o bază de date de idei utilizând hipertextul. În modelul lui Berners-Lee, baza de date este formată din atâtea baze de date câte sunt necesare pe atâtea calculatoare câte sunt necesare. Oricine ar putea să-și actualizeze baza de date și să furnizeze rezultatele obținute pe calculator. Toți ar putea să aibă acces liber la toate datele și să creeze relații prin conexiuni și marcarea conexiunilor. Nu ar exista nici o diferență între discipline diferite; cunoștințele interdiciplinare ar fi disponibile oricând, putând fi ușor de căutat și vizualizat. Crearea legăturilor între discipline ar fi la fel de ușoară ca stabilirea unei conexiuni.

O dată cu dezvoltarea acestei idei, Berners-Lee a făcut de asemenea alte câteva dezvoltări care au dus la World Wide Web-ul pe care îl știm. Mai întâi și-a dat seama că un sistem ierarhic de informații s-ar prăbuși sub propria-i greutate. El a observat de asemenea că informația pe un sistem nu ar fi limitată la text și că indexarea ar limita accesul la informație și ar ascunde-o de aceia care au nevoie de date dintr-un context diferit. Răspunsul său la aceste probleme a constat în hipertextul distribuit, în care oamenii își păstrau datele pe calculatoarele lor, acestea fiind legate într-o rețea unde datele erau diponibile publicului care accesa acea rețea.

Berners-Lee a scris software-ul serverului și al clientului pe calculatorul său, încercând apoi distribuirea printre colegii săi. Împreună cu aceștia a dus o luptă asiduă în vederea depășirii protocoalelor și a sfârșit prin crearea adreselor URL (Uniform Resource Locator), a protocolului HTTP (Hypertext Transport Protocol) și a limbajului HTML.

Crearea HTML-ului

Când Tim Berners-Lee a fost pus față în față cu mai multe opțiuni de implementare a conceptelor sale, el a ales SGML (Standard Generalized Markup Language). Această decizie a condus la independența Web-ului față de orice sistem de operare sau limbaj individual. De asemenea el a pregătit terenul pentru dezbaterea în curs asupra menirii HTML-ului.

SGML-ul este un standard recunoscut internațional pentru procesarea informației sub formă de text, care furnizează o modalitate organizată a structurării conținutului pentru o utilizare largă. Documentele marcate în conformitate cu regulile conturate în SGML au două elemente de bază – conținutul și informația despre conținut.

Acest mod de abordare structuralizat a fost ideal pentru scopul lui Berners-Lee. Un om de știință putea crea conținutul, marcajul SGML putea fi adăugat mai târziu și documentul rezultat putea fi făcut disponibil prin rețea oricui avea acces la aceasta. Sună cunoscut?

SGML este o platformă independentă, necesitând doar un calculator și software-ul adecvat pentru a analiza documentele astfel încât datele să aibă un înțeles. Programul de analizare a fost oarecum personalizat, astfel încât fiecare utilizator să profite de mașina pe care o folosea la acel moment. Depindea de programul de parcurgere să afișeze textul de marcaj într-o manieră corespunzătoare platformei – astfel încât un Heading 1 era diferit de un Heading 2 și Emphasis arăta diferit față de Strong. Deoarece marcajul identifica doar natura conținutului, era responsabilitatea cititorului de a decide cum să reprezinte niveluri particulare de înțeles pentru platformă.

Înaintea HTML-ului, unui autor de documente nu trebuia să-i pese de modul în care un document apărea pe monitorul cuiva. Era acceptat faptul că înfățișarea era domeniul utilizatorului. Autorii HTML-ului nu puteau specifica nume de fonturi sau mărimi, margini, spații albe sau orice altă trăsătură grafică.

SCURTĂ TEORIE A MARCAJULUI DE HIPERTEXT

Autorul unui sistem de informație neliniară trebuie să ofere o metodă pentru legarea ideilor înrudite și un mecanism de căutare pentru probleme specifice care s-ar putea să nu aibă răspunsuri evidente. În fine, un sistem de informație neliniară nu poate avea limite. Cu un mecanism care leagă uniform, cititorul nu este restricționat la lucrarea unui autor, ci poate urma legături pentru a găsi documente oriunde în lume. Aceste beneficii pot fi pe de altă parte dezavantaje dacă sunt prost implementate. Menținerea contextului într-o lume a hipertextului este dificilă, dar necesară.

Acestea fiind spuse, să vedem care este baza lumii hipertextului, pe care noi o numim World Wide Web. Pentru a pregăti hipertextul, informația este secționată în unități mici, numite *noduri*. Apoi în noduri sunt înglobate *hiperlegăturile* ce conduc la noduri diferite. Procesul de navigare printre noduri utilizând legăturile acestora este numit *browsing* (răsfoire). O colecție de noduri interconectate de hiperlegături se numește *web* (pânză de păianjen), ceea ce face World Wide Web-ul un sistem global de hipertext.

Designul efectiv pentru documentele cu hipertext ce fac parte dintr-un sistem neliniar este mult mai dificil decât acela pentru un text narativ. Această problemă a condus la numeroase metode care sunt folositoare pentru designul paginilor de Web. O metodă este divizarea textului în fragmente care tratează o temă, un subiect sau o idee. După ce materialul este secționat, trebuie scris în așa fel încât să poată fi citit ca un nod de sine stătător de către un utilizator care poate să fi ajuns la acel nod prin oricare dintre legăturile oferite de autor sau printr-un motor de căutare.

Autorul nu poate presupune că utilizatorul are vreo informație din pasajele „anterioare", așa cum este în cazul scrierilor narative. Nodurile prost secționate pot duce la o succedare de bucățele de informație disparate și cu conținut redus sau la un nod care este o colecție de legături ilizibilă ce sar în ochi și copleșesc conținutul.

Textul liniar își dezvoltă conținutul din tot ceea ce a intervenit înainte. În sistemele de informație neliniară, cum este Web-ul, cuvinte ca *următor* sau *precedent* devin capcane de context pentru autor. Într-o situație de hipertext, „următorul" s-ar putea să nu apară niciodată, iar „precedentul" s-ar putea să nu fi existat. Un cititor poate urma chiar o legătură către un alt nod înainte să termine nodul curent și să nu se mai întoarcă niciodată la nodul originar.

Fiecare nod trebuie nu numai să fie corect secționat, ci și să fie scris astfel încât să furnizeze un set de indicii de context pe care un cititor fără nici o legătură anterioară să îl poată folosi pentru a se orienta către informația prezentată.

(continuare)

(continuare)
O opțiune este să furnizeze legături către o ierarhie unde cititorii se pot orienta singuri, dacă trebuie, și să se conecteze la alte noduri relevante. În plus, conexiunile din interiorul documentului trebuie să aibă un scop care să fie evident celor care intră în nod din alte situri Web.

Evoluția HTML-ului

Una dintre cerințele pentru a crea un sistem de informație neliniară utilizabil este de a-l face ușor de înțeles și utilizat de către neexperimentați. Spre deosebire de fișierele AppleScript, HyperTalk și WinHelp, marcarea textului pentru prezentarea pe Web folosește un set limitat de etichete, care sunt simple și ușor de înțeles. Această simplicitate a fost greu de găsit și dispare rapid sub atacul unor noi atribute pentru interfețe grafice.

Nivelul 0 HTML

Folosind SGML-ul pentru a descrie cum se formatează conținutul, Berners-Lee a folosit un alt aspect al SGML-ului – abilitatea de a defini alte limbaje. Berners-Lee și colegii săi au folosit SGML-ul pentru a descrie regulile limbajului de marcare a hipertextului într-o definiție a tipului de document (DTD), baza structurii documentelor pe World Wide Web. Despre DTD-uri se discută mai detaliat în Capitolul 4, „Definiția tipului de document HTML".

În prima sa implementare, HTML-ul a urmat regulile SGML-ului de marcaj al conținutului independent de platformă fără prevederi de reprezentare a documentului în HTML-ul însuși. Utilizatorii trebuiau să ofere formatarea prin combinația lor browser-platformă (clientul), iar clientul nu era cunoscut la momentul creării documentului.

HTML-ul era la Nivelul 0 la prima sa implementare, în 1990. La acel moment, mijloacele de comunicație de pe Internet includeau e-mail, FTP (protocolul de transport al fișierelor) și Telnet, toate folosind TCP/IP (transmission control protocol/Internet protocol – protocol de control al transmisiei/protocol Internet). Gopher devenea un mod popular de indexare a informației și serverele Gopher erau introduse pe Internet o dată cu Archie, Veronica și Jughead. WAIS (Wide Area Information Server – Server de informație pe arii extinse) era de asemenea introdus ca o modalitate de furnizare a accesului, în special pentru studenții din campusurile colegiilor. Pentru a permite în lumea întreagă accesul la documentele HTML, Berners-Lee și asociații săi au introdus ideea HTTP-ului și a adreselor URL pentru furnizarea adreselor și a mijloacelor de localizare a datelor.

La Nivelul 0, HTML-ul oferea mijloace independente de platformă pentru marcarea datelor pentru interschimb. Conceptul era acela că serverele aveau să stocheze și să furnizeze date, iar clienții aveau să le recupereze și să le afișeze. De aceea, nodurile HTML puteau oferi acces nu numai către alte noduri HTML, ci și către spații Gopher, FTP, Network News și așa mai departe.

HTML 0.0 era foarte apropiat de SGML. Singurul element solicitat era TITLE, și multe pagini mai vechi încep cu un titlu și apoi intră direct în text. Nu există HTML, nici HEAD, nici BODY. O etichetă <P> era deseori utilizată la sfârșitul fiecărui paragraf pentru a separa paragrafele între ele. Existau șase niveluri de antet, dar se aștepta ca fiecare nivel să fie folosit numai o dată și nivelurile să fie utilizate succesiv.

Nivelul 1 permitea o etichetă <BODY> și autorii puteau include adrese, ancore, ghilimele, întreruperi de linie, anteturi, imagini, liste, paragrafe și text preformatat.

Nivelul 1 HTML

Discuțiile asupra HTML 0 au adus la lumină mai multe deficiențe, așa că în 1992 Dan Connolly a început să lucreze la HTML 1.0, care a fost lansat pe Internet în același an. A fost adăugată ideea unui container HTML, cu un element HEAD separat de elementul BODY. Unele elemente aveau nevoie de eticheta de deschidere și închidere.

Alături de TITLE, elementul HEAD putea conține atribute ca ISINDEX, LINK și BASE, dând documentului un context în interiorul unui univers mai larg. Ca și elementul ANCHOR, elementul IMAGE permitea ca fișierele GIF (format de interschimb grafic) să fie afișate în interiorul textului, dacă browserul suporta acest format. Era de asemenea introdusă linia orizontală, începând drumul periculos de a da autorului control asupra imaginii finale a documentului în fața utilizatorului.

Nivelul 1 introducea de asemenea formulare, care făceau posibilă pentru autori introducerea câmpurilor pe nodurile lor, care permiteau feedback-ul de la utilizatori și deschideau ușa către o interacțiune considerabilă prin scriptarea Common Gateway Interface (CGI). Potențialul formularelor și scripturile CGI evoluează în continuare, chiar cu adăugarea unor noi capacități de formulare în HTML 4 (vezi Capitolul 15, „Crearea și utilizarea formularelor HTML" și Capitolul 25, „Să înțelegem interfața CGI").

Alte ethichete introduse o dată cu Nivelul 1 includeau elementele CITE și CODE, alături de elemente de stil care sugerau utilizarea și nu aspectul: KBD (tastatura), SAMP (ieșire de model), STRONG (subliniere puternică) și VAR (argument de variabilă sau de program). De asemenea erau incluse elemente pentru B (bold), I (italic) și TT (monospațiat). În timp ce Nivelul 1 era subiect de discuție, erau propuse elemente adiționale și capacități pentru versiunea HTML+, care se pregătea să intre în scenă.

HTML+ și progresul graficii

Un alt pionier timpuriu al Web-ului, Dave Raggett, a propus o altă extensie a HTML-ului, denumită HTML+, care încorpora elemente grafice și de afișare în HTML. Prin încorporarea acestor elemente, HTML+ putea profita de capacitățile Mosaic-ului și ale altor browsere bazate pe grafică care deveneau disponibile. HTML+ oferea mijloace de conectare la documente PostScript, imagini JPEG, filme MPEG și sunete și de accesare a acestora. MIME (Multipurpose Internet Mail Extensions) era folosit pentru a permite extensia tipurilor de fișiere HTML; datele aflate la dispoziție pe Internet nu erau restricționate de convenția de 7 biți ASCII a Internetului. Această propunere făcea să dipară limitele dintre tipurile de date aflate pe Internet. Datorită dezvoltărilor anterioare, nu exista nici o limitare asupra numărului de calculatoare, a numărului de utilizatori sau a cantității sau tipului de date.

HTML+ a fost un important pas înainte pentru evoluția HTML-ului. Raggett a inclus elemente și atribute pentru superscripturi și subscripturi, note de subsol, margini, text inserat sau șters, alinierea conținutului, anexe, tabele, formule matematice, seturi extinse de entități și figuri.

Cu HTML+, Raggett a căutat să extindă HTML-ul dincolo de conceptul unui articol cu hiperlegături aflat pe un calculator. El a introdus HTML+ ca metodă de conectare dincolo de text, la fișiere de sunet, imagini animate, PostScript și oricare alt format introdus cu extensiile MIME.

Chiar și cu introducerea elementelor pentru formatare vizuală și control, Raggett nu se ocupa cu instrucțiuni despre aspect și își păstra propunerea în limitele constrângerilor SGML-ului. Elementele, atributele și valorile continuau să nu facă deosebirea între literele mari și cele mici, în conformitate cu regulile SGML-ului. Analizatorii (parsers) HTML+ ignorau etichetele pe care nu le recunoșteau, permițând autorilor și utilizatorilor să folosească alt marcaj SGML în documentele HTML pentru indexare și alte procesări, din afara HTML-ului. Pentru a asigura detectarea HTML-ului de către browsere și interpretarea corectă a acestuia, a fost sugerată folosirea unui identificator de documente, care avea să preceadă orice altă informație din document, inclusiv eticheta <HTML>. Acest identificator face în continuare parte din standardul HTML, apărând sub forma etichetei <!DOCTYPE>.

HTML 2

Pretențiile în creștere asupra HTML-ului au condus consorțiul World Wide Web, nou format, (W3C) la introducerea versiunii HTML 2 (vezi secțiunea „Consorțiul World Wide Web" în continuare în acest capitol pentru mai multe informații despre W3C). Nivelul 2 folosea elementul FORM cu INPUT, SELECT, OPTION și TEXTAREA, plus elementul BR pentru întreruperi de linie.

HTML 2 adăuga de asemenea elementul META pentru descrierea detaliată a documentului și o cale pentru indexarea și catalogarea conținutului. De asemenea schimba descrierile secțiunilor head și body și ale elementelor ancoră, bază, listă, imagine, legătură și titlu.

Grupul din W3C pentru dezvoltarea versiunii HTML 2 a creat o specificație care asigura compatibilitatea standardelor existente HTML. DTD oferea de asemenea etichete de container opționale, cum ar fi eticheta <P>.

HTML 2 nu a descoperit nimic nou în ceea ce privește cpacitățile HTML. Mai degrabă, a continuat și a făcut oficial status quo-ul. Era practicabil de catre toți agenții utilizatorilor, pentru că urma regulile stabilite de SGML și nu necesita dezbateri extensive pentru că nu exista nici o controversă dacă elementele propuse erau necesare.

SUGESTIE

Veți vedea termenul *agent al utilizatorului* ocazional în această carte. Pentru aceste discuții, puteți considera *agentul utilizatorului* ca fiind sinonim cu *browser*. În realitate, un agent este orice aplicație ce poate procesa un document HTML.

O dată cu reglementările oficiale introduse de W3C, lumea avea acum un standard HTML care să servească ca etalon față de care puteau fi comparate browserele și marcajul.

HTML 3

HTML 3 a fost propus ca o încercare de a răspunde cerințelor contradictorii pentru un limbaj de marcare care să funcționeze pe toate platformele și pentru un limbaj de descriere a paginii care să fie acceptat de alte companii de sofware, pe lângă Netscape. Teoria era că HTML 3 va fi în totalitate SGML, dar va permite sugestii pentru browsere asupra modului în care să afișeze un anume text. HTML 3 a fost dezvoltat sub administrația lui Dave Raggett, care dăduse de asemenea imboldul aflat în spatele HTML+ și contribuise la HTML 2.

Elementele propuse pentru HTML 3 includeau elementul FIG, care suporta curgerea textului în jurul figurilor. A fost de asemenea sugerat suportul pentru ecuațiile matematice și un element TABLE pentru a ajuta la aranjarea datelor în formă de tabel fără șicana provocată de elementul PRE.

Atributul ALIGN a fost adăugat mai multor elemente, inclusiv IMG, P și HR, permițând autorilor să folosească alinierea la stânga, la dreapta sau pe centru. La nivelul caracterelor, HTML 3 propunea de asemenea noi elemente logice, incluzând etichete pentru definiții, citate, limbaj, text inserat și șters. Unele dintre etichetele fizice propuse erau text subliniat, text mai mare sau mai mic, subscript și superscript. Atribute adiționale erau de asemenea propuse pentru imagini de fundal, anexe, note de subsol și manșete.

A fost de asemenea acordată considerație foilor de stil în HTML 3. Prin îndepărtarea majorității elementelor orientate către afișaj din HTML și prin plasarea acestora în foi de stil, putea fi păstrată puritatea unui limbaj SGML, oferind totuși o modalitate autorilor de a încorpora designul grafic pe Web.

Au fost scrise mai multe cărți despre trecerea de la Nivelul 2 la Nivelul 3 și despre pericolele utilizării unui limbaj de marcare neaprobat ca standard încă. HTML 2 a existat ca standard, dar HTML 3 a lâncezit în procesul de standardizare, în timp ce Microsoft și Netscape se mișcau rapid înainte cu dezvoltări ale browserelor. Atunci când schița HTML 3 expirase, Microsoft își introdusese propriul Internet Explorer, iar Netscape Navigator avansase la o altă versiune – și nici unul dintre aceste browsere nu mai era limitat la standardul curent sau la cel propus. Datorită tuturor acestor forme disponibile de HTML, domneau confuzia și iritarea.

HTML 3.2

La urma urmelor, HTML 3 era prea ambițios. Așa că a fost proiectat HTML 3.2 ca o recunoaștere a realității și a încorporat multe dintre etichetele deja aflate în uz pe paginile de Web. A adăugat etichetele SCRIPT și STYLE pentru a crea mai mult loc pentru limbajele de scriptare și pentru foile de stil. A oficializat practici precum culorile pentru fundal, text, legături și mărime, aliniere și spațieri pentru imagini.

HTML 3.2 era noul standard, înlocuind HTML 2. Acesta oferea elemente și atribute noi, care înveseleau paginile Web cu animație, culori și sunet. A făcut posibilă crearea unor pagini Web atractive și dinamice, care se supuneau unui standard stabilit. Existau neînțelegeri asupra celor care fixau standardele, dar cel puțin un DTD recunoscut oficial era acum disponibil.

HTML 4

Ultima versiune a HTML-ului folosește unele idei și modificări propuse pentru HTML 3, care au fost abandonate la apariția HTML 3.2. HTML 4 propune separarea stilurilor fizice de marcarea conținutului printr-o folosire mai intensă a foilor de stil. Mulți văd soluția foilor de stil ca pe cea mai bună metodă de a sugera aspectul documentului și totuși de a permite utilizarea completă a HTML-ului ca o metodă compatibilă SGML de traversare a prăpastiei dintre utilizatori și sistemele de operare.

HTML 4 folosește, de asemenea, unora dintre scopurile HTML-ului 3.2 prin confirmarea practicii curente. Elementul FRAME este acum definit oficial altundeva decât în produsele firmelor Netscape sau Microsoft. Pentru a direcționa interesele competitive dintre mini-aplicații, plug-ins-uri și controale ActiveX, W3C a introdus elementul OBJECT, care beneficiază de destulă flexibilitate pentru a servi în alte roluri în viitor.

Noul standard pentru HTML utilizează de asemenea și alte propuneri ale W3C pentru Web. În loc să încorporeze toate funcționalitățile unei pagini Web într-un singur standard, HTML 4 lucrează mai degrabă cu interfețe către multe alte standarde. Standardul HTML 4 include elementele STYLE, DIV și SPAN pentru încorporarea foilor de stil, dar foaia reală de stil este o entitate separată, ca în Cascading Style Sheets 1 (Foi de stil în cascadă 1).

Consorțiul World Wide Web

În timpul unui seminar din iulie 1993, Berners-Lee a propus fondarea unui consorțiu care să se ocupe de problemele legate de stabilirea standardelor pentru HTML și HTTP. El împreună cu angajații săi nu formau un grup potrivit pentru a menține standardele, astfel încât a fost sugerat un consorțiu pentru stabilirea unui model de referință pentru standarde, propuneri de teste pentru determinarea normelor și acordarea aprobării oficiale.

Carta W3C

Ca rezultat al propunerii lui Berners-Lee, s-a format în 1994 Consorțiul World Wide Web (W3C) pentru a încerca să aducă ceva din ordinea foarte necesară în lumea HTML. Fondat ca un consorțiu industrial, scopul lui W3C este dezvoltarea unor standarde de adoptare comune; o dată adoptate, acele standarde urmează să fie folosite de membrii consorțiului

în software-ul lor, fie el server sau client. W3C încearcă de asemenea să dezvolte protocoale Web mai rapide și mai eficiente pentru imagini, sunete și video.

Lucrările W3C

Cum își face W3C treaba? Primul pas este identificarea unei probleme sau a unei tehnologii ce afectează World Wide Web-ul. Organul de conducere al W3C, numit Advisory Committee, desemnează un grup de lucru care este format din experți adunați pentru a lucra împreună în vederea rezolvării unei probleme tehnice specifice și bine definite, cum ar fi standardele HTML.

Grupul de lucru dezvoltă o serie de *schițe de lucru* (working drafts) până când ajung la un consens și un acord stabil. Apoi, dacă directorul W3C aprobă, schița este avansată la titlul de *Proposed Recommendation* (Recomandare propusă), de asemenea denumită *Draft Recommendation* (Recomandare de schiță), care este trimisă tuturor membrilor W3C pentru a fi discutată. În funcție de numărul și natura comentariilor, directorul W3C poate emite rezultatul ca o *Recomandare* (fie exact cum era, fie cu câteva modificări minore), poate să o returneze grupului de lucru sau chiar să dispună renunțarea la ea.

Dacă vă consolează cu ceva, W3C nu lucrează fără să țină seama de domeniul de activitate pe care îl afectează. W3C este găzduit de Institutul de Tehnologie Massachusetts, dar membrii comitetului sunt reprezentanți ai companiilor World Wide Web, inclusiv ai organizațiilor precum Microsoft, Netscape, Sun Microsystems, IBM și altele. Deși costă foarte mult ca o organizație să devină membru al W3C, nici o organizație sau companie nu are mai multă influență asupra recomandării finale a W3C decât o alta.

Cu Netscape și Microsoft ca membri ai săi, W3C a devenit terenul de luptă pentru inima și sufletul Web-ului. Carta consorțiului – de a aduce ordine în dezvoltarea protocoalelor HTML și HTTP – a cedat locul intereselor comerciale și afirmării unora dintre membrii consorțiului. Deși mulți au prezis că aducerea lui Netscape în peisaj va forța compania să renunțe la acțiunile sale rebele, valul comercial inundă în mod implacabil sala de conferință. Deși se încearcă eliberarea standardelor de sub influența vânzătorilor de software, noile standarde HTML includ în continuare articole care se înclină în fața realității etichetelor de proprietar introduse de Microsoft și Netscape, cum ar fi cadrele.

Rezumat

O combinație de trei factori l-a făcut pe Berners-Lee creatorul Web-ului. În primul rând, el a lucrat într-o perioadă în care calculatoarele erau suficient de puternice, suficient de mici și suficient de ieftine pentru a fi utilizate ca o infrastructură a World Wide Web-ului. Aceasta este o întâmplare pe care Berners-Lee nu ar fi putut să o dirijeze.

Al doilea factor a fost abilitatea sa de a vedea dincolo de o necesitate specifică și o soluție anume, să sesizeze posibilitatea unei realizări importante, ce apărea într-un moment favorabil și dispunea de uneltele necesare – un mod eficient de a stoca și recupera informațiile neliniare. Al treilea factor a fost norocul: Berners-Lee s-a aflat la locul potrivit în momentul potrivit, având uneltele ce îi lipsiseră lui Vannevar Bush.

Berners-Lee a folosit uneltele pe care le avea la îndemână, a inventat ceea ce nu a avut, a scris software-ul, a schițat propuneri și a coordonat eforturile multor oameni pentru a-și transforma ideea în practică. Restul, după cum se spune, este istorie.

Capitolul 3
Structura unui document HTML

de Rick Darnell

În acest capitol:

- **Structura unui document 30**
- **Structura unei secțiuni 31**
- **Infrastructura altor elemente 33**

În limbajul de marcare a hipertextului totul se rezumă la structură. Este adevărat că a existat o invazie de materiale orientate către înfățișare, dar în esență HTML se referă la definirea structurii unui document.

La urma urmelor, un limbaj de marcare este folosit pentru a descrie care este rolul fiecărei părți din document în raport cu celelalte. Poate indica una dintre părți ca fiind un titlu de nivelul trei, că secțiunea de dedesubt este un paragraf și că ultima parte este o adresă. În cadrul paragrafului, puteți marca citate, acronime și orice altă bucată din text. Rezultatul este un document în cadrul căruia puteți vedea *scopul* fiecărei părți a textului, în locul înfățișării intenționate.

În contrast, programele de editare cu care sunt familiarizați majoritatea utilizatorilor (Microsoft Word și Publisher, Aldus Pagemaker) pun accentul pe înfățișarea documentului și nu pe structura sa. În loc să indice scopul textului, autorii selectează direct tipul caracterelor, culorile, marginile și chenarele pentru a controla cu precizie cum va arăta în final pagina.

Acest capitol aruncă o privire asupra aspectului structural al unui document HTML, începând cu părțile mari și aprofundând apoi detaliile. Aspectul unui document HTML este controlat cu foile de stil, care sunt menționate în acest capitol ca o parte componentă a structurii. Explicații detaliate și exemple sunt găsite în Partea a IV-a, „Tehnici de prezentare", și în Partea a VII-a, „Designul efectiv al unei pagini Web".

TERMINOLOGIA HTML

Înainte de a începe să vorbim despre structura HTML-ului și subiectele legate de aceasta, este important să stabilim o bază comună în privința termenilor. Ceea ce urmează este o listă cu definițiile termenilor care sunt folosiți în acest capitol și în această carte.

- **Atribut:** Un atribut furnizează instrucțiuni adiționale browserului despre o etichetă. Informațiile variază de la etichetă la etichetă și pot include subiecte cum ar fi locația fișierelor, mărimea, numele sau stilul lor.
- **Browser:** Un browser de Web este un motor de parcurgere (parsing) special care evaluează etichetele și conținutul unui fișier HTML, pe care îl afișează în concordanță cu posibilitățile și regulile platformei și capacităților sale. Ca exemplu, un browser numai pentru text afișează textul alternativ în locul unei imagini, pe când un browser grafic afișează însăși imaginea. Un browser pentru o platformă monocromă ignoră setările culorilor din foaia de stil sau redă culorile ca nuanțe de gri.
- **DTD (Document Type Definition – definirea tipului de document):** O colecție de reguli scrise în SGML (Standard Generalized Markup Language – limbajul de marcare standard generalizat) care definesc sintaxa și structura unui anume tip de document, cum ar fi HTML (vezi Capitolul 4, „Definiția tipului de document HTML")

(continuare)

(continuare)

- **Element:** O componentă distinctivă a structurii unui document, cum ar fi titlul, un paragraf sau o listă. Când ne referim la forma sa aplicată în cadrul unui document, un element se mai numește *etichetă* (tag). În această carte, termenul *element* este sinonim cu *etichetă*.
- **Etichetă:** Un cod care identifică un element pentru ca browserul sau alt program de parcurgere să știe în ce mod să afișeze conținutul. Etichetele sunt încadrate de caractere de delimitare (paranteze ascuțite). În această carte, termenul *etichetă* (tag) este sinonim cu *element*.

Structura unui document HTML are trei niveluri de bază. Primul este structura documentului, unde documentul este împărțit în secțiuni mari în funcție de scopul lor, cum ar fi antet, corp și script.

Al doilea este divizarea în cadrul secțiunilor mari. Acest nivel se aplică în primul rând antetului și corpului, deoarece fiecare are propriul grup de conținut, cum ar fi titluri, metadate, diviziuni, paragrafe, formulare și titluri de secțiuni.

Al treilea nivel există în primul rând în corp. Este infrastructura elementelor și a copiilor lor, cum ar fi liste și liste de obiecte, formulare și elemente de formular și tabele cu rânduri și celule de date.

Veți explora fiecare dintre aceste secțiuni majore, scopurile, precum și conținutul lor în acest capitol.

Structura unui document

Fiecare document HTML, încă de la creare, are două părți principale: un antet și un corp. Chiar dacă nu sunt special delimitate, HTML presupune existența lor. HTML oferă de asemenea șansa de a adăuga și alte componente funcționale paginii, sub forma foilor de stil, scripturilor și seturilor de cadre.

Dar acestea vor fi discutate mai târziu. Deși fac parte din definirea unui document HTML, două etichete există în afara structurii documentului: `<!DOCTYPE>` și `<HTML>`. Prima identifică versiunea de HTML folosită pentru pagina respectivă și regulile pe care aceasta le respectă. A doua etichetă identifică un limbaj global și direcția textului pentru documentul respectiv, în plus față de a marca începutul și sfârșitul conținutului HTML. Amândouă elementele sunt explicate detaliat în Capitolul 6, „Structurarea textului".

Cu aceste două etichete puteți crea un document HTML cu următoarele trei linii:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0//EN">
<HTML>
</HTML>
```

Nu va arăta cine știe ce pe un browser. De fapt, nu va afișa nimic. Totuși, un browser care se conformează standardelor HTML o să vadă automat documentul, de parcă ar fi fost scris sub forma:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0//EN">
<HTML>
<HEAD>
</HEAD>
<BODY>
</BODY>
</HTML>
```

După cum spuneam, fiecare document HTML se presupune că are un antet și un corp, chiar dacă nu sunt delimitate explicit.

SUGESTIE

În acest moment, vă recomand să nu vă bazați pe browser pentru a defini începutul și sfârșitul antetului și corpului. Ar trebui să introduceți dumneavoastră aceste elemente ca parte din fiecare document pe care îl creați. Este o opțiune de profesionist, care economisește timp și efort în reviziile și editările ulterioare.

Elementul `HEAD` include o gamă de informații pentru browser, inclusiv un titlu pentru document, cuvinte cheie, informații de legătură și alte informații cu care utilizatorul nu trebuie să lucreze direct. Pentru mai multe informații despre cum să compuneți elementul `HEAD`, vezi Capitolul 6.

`BODY` este ceea ce vede utilizatorul în momentul în care documentul este redat de browser. Include text și obiecte, cum ar fi imagini și miniaplicații, la un loc cu etichete de marcare pentru a defini structura lor precum și relația dintre ele.

Mai sunt trei componente opționale în structura unui document HTML: foile de stil, scripturile și seturile de cadre. Aceste elemente trebuie definite explicit sau browserul nu le va recunoaște existența.

O foaie de stil, creată cu elementul `STYLE`, este un lucru nou în HTML. Vă oferă un nou mod pentru a defini felul în care va arăta documentul, inclusiv familia de caractere și fontul, culorile, chenarele, marginile și alinierea. În trecut, aceste setări se făceau adăugând etichete direct în corp. Prin adăugarea foilor de stil ca un element de structură al documentului, înfățișarea și conținutul sunt separate pentru a da browserelor posibilitatea unui control mai precis. Foile de stil sunt discutate mult mai detaliat în Capitolele 18, „Utilizarea foilor de stil în cascadă", 19, „Foile de stil în cascadă și poziționarea elementelor" și 21, „Folosirea foilor de stil JavaScript".

Elementul `SCRIPT` încadrează o secțiune de cod de programare inclusă direct în pagina de Web. Acest element oferă utilizatorului o modalitate de a interacționa direct cu un document, fără a se baza pe ajutorul unui server Web. Un document poate include mai multe scripturi în mai multe limbaje. Alegerile obișnuite la ora actuală pentru limbaje de scriptare sunt JavaScript (Capitolul 23, „Scurtă privire asupra limbajului JavaScript") și VBScript (Capitolul 24„Folosirea limbajului VBScript"). Alte opțiuni mai pot fi Python și Perl.

Ultimul element opțional este `FRAMESET`, care dă instrucțiuni browserului cum să subdivizeze fereastra într-un set de ferestre mai mici, fiecare cu propriul său conținut. În esență, fiecare fereastră devine un „minibrowser" care poate găzdui un nou document HTML. Elementul `FRAMESET` este strâns legat de elementul `BODY`. În browserele care suportă cadrele elementul `FRAMESET` înlocuiește elementul `BODY`. În browserele care nu suportă cadrele `FRAMESET` este ignorat și este folosit elemntul `BODY`. Implementarea seturilor de cadre este discutată în Capitolul 20, „Crearea cadrelor".

Structura unei secțiuni

Fiecare element major din HTML are propria sa structură și necesități, reflectate în scopul definit pentru elementul respectiv. Două din elementele de document prezentate în secțiunea anterioară, `STYLE` și `SCRIPT`, sunt unice prin faptul că structura depinde de conținutul lor.

În cazul elementului `STYLE`, formatarea, sintaxa și definirea stilului variază în funcție de alegerea autorului: foi de stil în cascadă (Capitolul 18) sau foi de stil JavaScript (Capitolul 21). Pentru `SCRIPT` se aplică aceleași excepții, în funcție de ce limbaj de scriptare

este folosit. Un script poate conține o funcție, mai multe funcții sau poate fi o aplicație de sine stătătoare.

Structurile HEAD și BODY

Structurile antetului și corpului unui document HTML au o trăsătură comună: nu se specifică o machetă sau o ordine anume. Diferența constă în faptul că elementele antetului există ca frați, în timp ce corpul poate crea o ierarhie de părinți și copii.

Amintiți-vă că antetul conține informații despre document care pot afecta înfățișarea corpului, dar care nu sunt afișate în mod obligatoriu utilizatorului. Nu există în antet elemente care să fie mai importante decât celelalte – TITLE este la fel de important ca și LINK sau META. Ele sunt tratate în mod diferit doar conform scopului lor.

Pe de altă parte, este elementul BODY. În mod obișnuit conține o ierarhie de elemente. Luați în considerare acest conținut simplu al unui corp:

```
<BODY>
<H1>Packing a first aid kit</H1>
<P>When you pack a first aid kit, you don't need to bring everything for every
 possibility. Only concentrate on the probable, combined with enough supplies to
 match your skills for the possible.</P>
</BODY>
```

Corpul include doi copii, un element H1 și un element P. Dacă v-ați imaginat BODY ca pe un arbore, atunci cele două elemente ar ieși din trunchi în același loc. Puteți de asemenea imbrica elementele, după următorul model:

```
<P>When you pack a first aid kit, you don't need to bring everything for every
possibility. Only concentrate on the <EM>probable</EM>, combined with enough
supplies to match your skills for the possible.</P>
```

Elementul EM este o ramură a elementului P, care este o ramură a lui BODY. Când elementele sunt imbricate în corpul unui document HTML, între ele se stabilește o ierarhie care va varia pentru fiecare document în funcție de etichetele folosite de autor.

Structura FRAMESET

<FRAMESET> este un tip special de element de corp care, prin definiție, poate conține doar cadre și seturi de cadre. Conținutul său este mult mai controlat decât corpul. Totuși, ierarhia stabilită de plasarea unui <FRAMESET> se dezvoltă în același mod ca în cazul elementului <BODY>.

În liniile următoare fiecare dintre cele două elemente FRAME sunt frați și copii ai elementului FRAMESET:

```
<FRAMESET>
  <FRAME>
  <FRAME>
</FRAMESET>
```

Dacă adăugați alt set de cadre, cu altă configurație a cadrelor, obțineți ceva similar cu:

```
<FRAMESET>
  <FRAME>
  <FRAME>
  <FRAMESET>
      <FRAME>
  </FRAMESET>
</FRAMESET>
```


În acest caz sunt două elemente FRAME și un FRAMESET care sunt frați și copii ai primului FRAMESET. Fratele FRAMESET are și un FRAME ca un copil al său, fapt care îl transformă pe acesta în nepotul primului FRAMESET. HTML recunoaște acest tip de ierarhie atunci când este implementată în browsere și permite elementelor configurate așa să funcționeze împreună pe baza relației dintre ele.

Infrastructura altor elemente

În corpul unui document HTML, unele elemente au propria lor structură internă pentru a ajuta la definirea folosirii și înfățișării lor. Un exemplu bun pentru acest caz este elementul TABLE (discutat mai detaliat în Capitolul 9, "Organizarea datelor cu ajutorul tabelelor", și 10, „Macheta și prezentarea tabelelor").

Un tabel HTML este construit ca având cel puțin un rând și o celulă de date, precum și un document HTML are nevoie de cel puțin un antet și un corp:

```
<TABLE>
  <TR>
    <TD>Data
</TABLE>
```

Fiecare tabel poate avea o denumire descriptivă, un antet și conținutul unui corp:

```
<TABLE>
  <CAPTION> Caption </CAPTION>
  <THEAD>
    <TR>
      <TD>Data
  <TFOOT>
    <TR>
      <TD>Data
  <TBODY>
    <TR>
      <TD>Data
</TABLE>
```

Fiecare tabel devine un minidocument cu formă și structură proprii. Mai sunt și alte elemente asemănătoare cu TABLE în corp, inclusiv diferitele tipuri de liste (ordonate, neordonate, definiții), formulare și seturi de câmpuri.

Rezumat

La bază, HTML se referă doar la structură. Autorii dau un set de unelte pentru a marca scopul și intenția fiecărei bucăți din textul unei pagini Web. Apoi, totul depinde de browser pentru a decide cum să afișeze conținutul utilizatorului.

În ultimele două ediții ale HTML-ului (HTML 2.0 și 3.2), și-au creat drum în HTML etichetele nestructurale, printre care se numără etichetele pentru culoarea și mărimea fonturilor, pentru text aliniat, bold, italic sau subliniat. O dată cu HTML 4.0, etichetele și atributele de înfățișare nu mai sunt recomandate datorită foilor de stil, care reprezintă un mod de a controla înfățișarea fără folosirea directă a elementelor. Corpul HTML-ului revine numai la folosirea structurală.

Un document HTML este compus din părți aflate într-o ierarhie descendentă, începând de la antet, corp, foaie de stil, script și set de cadre. Antetul conține informații despre document care sunt implementate de browser sau folosite de motoare de căutare, iar

corpul include tot conținutul care se dorește a fi afișat utilizatorului. Foaia de stil definește modul în care documentul ar trebui să apară pe browser. Scriptul determină o metodă de a interacționa cu utilizatorul prin folosirea unui limbaj de programare simplificat. Setul de cadre împarte fereastra browserului într-un set de ferestre mai mici care pot conține, fiecare în parte, un document HTML.

Lucrând cu structurile separat de înfățișare, HTML se apropie de obținerea independenței complete față de cererile diferitelor tipuri de browsere și platforme.

Capitolul 4
Definiția tipului de document HTML

de Rick Darnell

În acest capitol:

- **SGML: Structura din spatele HTML-ului 35**
- **Declarația SGML-ului pentru HTML 4 37**
- **Definiția tipului de document pentru HTML 4 39**

În multe dintre exemplele prezentate în această carte veți observa o etichetă specială de marcare chiar la începutul fiecărui document, care arată cam așa:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0//EN">
```

Această linie uimitoare face referire la un document special, intitulat *Document Type Definition (DTD)*, care conține toate informațiile despre etichete, entități și structura globală a documentului. Este principalul proiect de execuție pentru fiecare document HTML.

În legătură cu DTD am o veste bună, dar și una proastă. Vestea bună: este disponibil gratuit pentru cercetare și examinare din partea Consorțiului World Wide Web (W3C). Vestea proastă: este mult prea lung pentru a fi discutat în totalitate în acest capitol. Așadar, voi discuta conceptele majore și caracteristicile sintaxei DTD-ului pentru HTML 4 prin prezentarea reală.din definiția originală. În acest mod, dumneavoastră veți înțelege întregul DTD fără a pierde prea mult timp trecând în revistă o groază de note tehnice.

SGML: Structura din spatele HTML-ului

SGML-ul este un standard internațional pentru descrierea textului marcat electronic. Acest lucru pare mult prea complicat, așa că îl voi face mai ușor de înțeles. SGML este o cale de a descrie un limbaj în termeni formali – limbaje de marcare, în particular.

Ceva similar cu SGML-ul este gramatica pe care ați învățat-o în școală. În timpul orelor de gramatică ați învățat despre substantive, verbe, subiecte, predicate și multe alte particularități ale limbii. Aceste lucruri nu reprezintă limba în sine – ele doar o descriu și ne arată cum se potrivesc părțile sale împreună.

Din punct de vedere istoric, cuvântul *marcare* descrie marcaje în cadrul textului pentru ca dactilografa, culegătorul sau altcineva să știe în ce mod trebuia să apară un anumit pasaj. Am văzut marcaje sub forma unor semne de editare, cum ar fi sublinierea cu linie ondulată pentru a reprezenta stilul bold, tăierea textului cu o linie pentru a indica ștergerea sau o singură subliniere pentru a indica stilul italic. Pe măsură ce s-au automatizat formatarea și tipărirea textului, termenul *marcare* a fost extins astfel încât să acopere toate tipurile de coduri speciale de marcare inserate în textele electronice pentru a controla formatarea, printarea și alte procesări.

Generalizând, *marcarea* se definește ca orice mijloc de a sugera o interpretare explicită a unei anumite părți din text. La cel mai mic nivel, cum ar fi documentele text ASCII, toate părțile printate conțin marcări: semnele de punctuație, literele mari și spațiile dintre cuvinte pot fi privite ca un fel de marcare. Aceste elemente de bază ajută cititorul să-și dea

seama când se termină un cuvânt și când începe altul, cum să interpreteze o propoziție (semnul exclamării sau al întrebării) sau unde începe și se sfârșește o subordonată.

Următorul nivel al acestui concept este *limbajul de marcare*: un set de convenții pentru marcare, independente de conținut, care codează un text sau altceva. Un limbaj de marcare trebuie să specifice cum se deosebește marcajul de conținut, ce marcaj este necesar, ce marcaj este permis și ce înseamnă acesta.

O SCURTĂ ISTORIE A SGML-ULUI

Rădăcinile SGML-ului se întorc în timp către sfârșitul anilor 1960, când conceptul de marcare descriptivă a văzut lumina zilei pentru întâia oară. După ce companiile începuseră să folosească calculatoare pentru procesarea documentelor, a devenit evident că un format de stocare ar trebui să aibă nu numai coduri de formatare interpretate de calculator, dar și informații descriptive, inteligibile pentru om, despre rolul fiecărui element conținut în document.

Primul sistem funcționabil care folosea aceste concepte a fost limbajul generalizat de marcare (GML) elaborat de IBM. Acest sistem original a fost predecesorul direct al SGML-ului și a furnizat prototipuri pentru multe dintre caracteristicile de bază ale acestuia, cum ar fi structura ierarhică a documentului și Definirea Tipului de Document. IBM a construit sisteme de publicare mainframe bazate pe GML, care au fost larg folosite pentru a produce documentație tehnică în cadrul corporațiilor.

În 1978, Institutul American Național de Standardizare (ANSI) a început cercetarea în domeniul marcării generale a documentului pentru a stabili un standard național pentru interschimbul informațiilor. Pentru aceasta au fost cooptați unii dintre cei ce au conceput GML-ul la IBM. Prima schiță ANSI a fost publicată în 1980 și s-a finalizat ca standard industrial în 1983.

În 1984, Organizația Internațională de Standardizare (ISO) s-a alăturat acestei acțiuni și a început pregătirea unei versiuni internaționale a standardului. Prima schiță a fost publicată în 1985, iar versiunea finală a fost emisă un an mai târziu sub numele *ISO 8879:1986 Information Processing-Text and Office Systems-Standard Generalized Markup Language*.

SGML este în general cel mai bine folosit în corporații și agenții mari, care produc multe documente și își pot permite să introducă un format unic pentru uz intern. În afară de IBM, SGML-ul mai este utilizat de Asociația Americană a Editurilor și Departamentul Apărării al Statelor Unite. În 1992, la Laboratorul European pentru Fizica Particulelor (CERN) a fost inițiat un alt proiect, pentru limbajul de marcare a hipertextului, dar aceasta este altă poveste, pe care o puteți găsi în Capitolul 2, "Ce este limbajul de marcare a hipertextului".

De ce să folosim SGML?

Sunt mai multe motive bune pentru a dezvolta un limbaj de marcare. În primul rând este necesar pentru a te concentra pe structura unui document și nu pe formatarea sa. Sistemul de marcare ar trebui să vă permită să construiți o ierarhie de etichete descriptive care nu numai că separă și descriu diferite părți din document, cum ar fi capitolele și paragrafele, dar și definesc o structură standardizată a documentului.

Altă cerință importantă a marcării este aceea de a permite foarte ușor extinderea și modificarea. Ideal, un utilizator ar trebui să poată defini un nou set de etichete dacă apare nevoia. În cele din urmă, marcarea nu ar trebui să fie caracteristică cuiva; ar trebui ca oricine să poată crea și folosi unelte de marcare bazate pe sistem și să producă programe implementând uneltele.

SGML, construit având în vedere cerințele marcajului, este strict descriptiv și nu are posibilitatea marcării vizuale a unui document sau a aspectelor orientate către design. Totuși, poate conlucra ușor cu sisteme procedurale de marcare externe, cum ar fi foile de stil.

SGML nu reprezintă un sistem de marcare, ci o metodă ordonată de creare a limbajelor de marcare pentru tipuri anume de documente. Sintaxa flexibilă a SGML-ului face posibilă construirea limbajelor de marcare, cum este HTML, care satisfac o mare varietate de cereri.

Definirea unei aplicații SGML

Un document creat cu SGML există ca o structură ierarhică de elemente imbricate. Un *element* poate fi un capitol, o secțiune, un paragraf, un tabel, un antet sau orice altă componentă a unui document. Deși pot fi anumite noțiuni comune în ceea ce privește aspectul acestora, SGML nu are nici un mijloc și nici o intenție să specifice vreun aspect al înfățișării lor. Autorul poate da indicații sau sugestii prin comentarii în cadrul aplicației SGML, dar acestea pot fi ignorate de agenții utilizatorilor sau motoarelor acestora de căutare.

Sunt patru mari părți într-o aplicație SGML:

- Prima este declarația SGML, care spune ce caractere sau entități sunt permise în aplicație.
- DTD prezintă sintaxa etichetelor de marcare. Acesta poate include și alte definiții, cum ar fi entități numerice sau literare. Toate documentele marcate cu aceeași ierarhie de elemente aparțin aceluiași *tip de document*. În loc să descrie un set de unelte pentru marcarea unui document, SGML definește structura unui anume tip de document prin intermediul DTD-ului.
- O specificație care descrie semantica marcării, ce nu poate fi exprimată prin DTD.
- Documentele în care se află conținutul și etichetele de marcare. Fiecare document trebuie să conțină o referire la DTD, care este folosită pentru a-l interpreta.

Primele trei părți sunt explicate în acest capitol, iar restul cărții acoperă crearea documentelor care corespund ultimei părți.

OBSERVAȚIE

Sintaxa SGML este foarte asemănătoare cu sintaxa HTML. Enunțurile SGML, ca și etichetele HTML, sunt încadrate de paranteze ascuțite (< >) și conțin un cuvânt cheie sau un nume urmat de unul sau mai mulți parametri separați de spații. Singura diferență majoră este că enunțurile SGML includ în mod tipic un semn de exclamație între paranteza de început și cuvântul cheie al declarației, după cum urmează:

```
<!ELEMENT IMG -O EMPTY -- Embedded image -->
```

Altă caracteristică a sintaxei SGML este folosirea comentariului în interiorul enunțurilor. Comentariul este ultimul lucru într-un enunț și începe și se termină cu două linii scurte (--).

Declarația SGML-ului pentru HTML 4

Declarația SGML este o construcție formală care specifică anumite informații generale despre o aplicație SGML și este asociată cu tipul documentului său. Include mai multe părți, printre care nivelul standardului SGML, setul de caractere, o estimare a necesarului de resurse ale sistemului, sintaxa, domeniul și caracteristicile.

Prima parte a unei declarații SGML este `SGML "ISO 8879:1986"`, care identifică versiunea ISO a SGML-ului caruia i se conformează declarația. În caz că nu ați observat, acesta este același standard original dezvoltat de ISO pe baza lucrărilor elaborate de ANSI și IBM.

După standard urmează un scurt comentariu care identifică declarația și folosirea acesteia, în acest caz, `SGML Declaration for HyperText Markup Language version 4.0`.

Urmează apoi secțiunea `CHARSET`, care specifică ce set de caractere să fie folosit de fiecare document, conform tipului documentului. Seturile de caractere se deosebesc prin caracte-

rele conținute și valorile folosite pentru a le reprezenta pe diferite calculatoare și sisteme d operare. Iată declarația CHARSET pentru HTML 4:

```
CHARSET
      BASESET "ISO Registration number 177//CHARSET
               ISO/IEC 10646-1:1993 UCS-4 with
               implementation level 3//ESC 2/5 2/15 4/6"
```

Declarația SGML definește exact ce set de caractere este folosit – valorile sau coduril permise într-un document care se conformă și ce caractere intenționează să reprezinte. Î loc să reinventeze roata, SGML folosește în mod obișnuit seturi de caractere care au fo deja adoptate de organizații de standardizare, cum ar fi ISO.

OBSERVAȚIE

Datorită faptului că HTML este folosit în toată lumea într-o varietate mare de limbi, trebuie să fie destul de flexibil încât să permită și alte caractere în afara celor latine. Acest lucru este obținut prin internaționalizarea HTML-ului, descrisă în recomandarea RFC 2070 (`ds.internic.net/rfc/rfc2070.txt`). RFC 2070 descrie un set extins de caractere bazat pe standardul Unicode, care folosește coduri pe 16 biți în loc de tradiționalele coduri pe 8 biți. Mai multe informații despre acest subiect puteți găsi în Capitolul 40, „Internaționalizarea limbajului HTML".

După CHARSET urmează secțiunea CAPACITY, care furnizează o estimare brută a resurselc sistemului necesare unui analizator SGML pentru a procesa DTD-ul. Sunt precizat mărimile diferitelor tipuri de memorie, după cum urmează:

```
CAPACITY      SGMLREF
              TOTALCAP        150000
              GRPCAP          150000
              ENTCAP          150000
```

Aceste informații nu sunt foarte sigure, deoarece folosirea memoriei depinde în mai măsură de arhitectura internă a programului de analizare. Majoritatea analizatoarelc SGML nu țin seama de aceste valori, iar cei care scriu declarația au grjă să atribuie numer suficient de mari pentru a se asigura că procesarea DTD-ului nu este întreruptă datorit faptului că o valoare a fost depășită.

După CAPACITY urmează o linie pentru declarația SCOPE:

```
SCOPE DOCUMENT
```

Scopul acestei linii este să specifice că următoarea secțiune a sintaxei este pentru a folosită atât în conformitate cu documentele SGML, cât și cu DTD-ul acestei aplicații.

Urmează apoi secțiunea SYNTAX, care definește caracteristicile sintaxei aplicației SGM cum ar fi prezentarea regulilor, a caracterelor de delimitare și control, precum și a numelc rezervate.

Secțiunea FEATURES a declarației SGML are parametri care controlează folosirea anumitc caracteristici ale sintaxei SGML. Sunt trei caracteristici SGML folosite în DTD-HTML-ului 4:

- MINIMIZE: Această categorie conține caracteristicile pentru minimizarea marcării, cu scopul de a facilita marcarea SGML și de a o face mai ușor de citit și de folosit de către utilizator. Caracteristicile de minimizare vă permit să evitați anumite etichete și alte instrucțiuni de marcare în situațiile în care contextul poate rezolva orice ambiguitate.

  Prima caracteristică pentru HTML este OMITTAG YES, pentru a permite anumitor elemente existe fără o etichetă de început sau de sfârșit. Cealaltă caracteristică este SHORTTAG YES, ca

nu este suportată de browserele cele mai folosite în prezent. Ea permite folosirea etichetelor goale, evitarea numelor atributelor și a altor ciudățenii tipografice, informația lipsă fiind furnizată de analizator prin niște reguli simple și eficace.

- LINK: Această categorie conține caracteristici care afectează procesarea elementelor atributelor. Ele sunt specifice SGML-ului și nu sunt permise în HTML.
- OTHER: Această ultimă categorie conține caracteristici variate care nu se încadrează în nici una din celelalte categorii. Caracteristica demnă de luat în seamă este FORMAL YES, care indică faptul că declarația entităților PUBLIC ar trebui să folosească sintaxa oficială a identificatorilor publici pentru a permite analizatorului să le substituie automat cu sursele externe.

Declarațiile SGML fiind analizate, este timpul să vă orientați atenția către ceea ce este interesant la o aplicație SGML – Definiția Tipului de Document.

Definiția Tipului de document pentru HTML 4

Definiția Tipului de Document pentru HTML 4 (HTML DTD) este prea lungă pentru a fi tipărită în întregime în acest capitol. Pentru aceia dintre dumneavoastră cărora le plac lucrurile atât de detaliate, actualul DTD poate fi găsit la `www.w3.org/TR/WD-html40-970708/HTML4.dtd`. În acest capitol, o să prezentăm bucăți din DTD, cu speranța că privind unele detalii veți putea să vă uitați la orice parte din întreg și să înțelegeți ce vedeți.

OBSERVAȚIE

În afară de versiunile consecutive de HTML, mai sunt câteva variante de HTML care deviază de la setul standard de caracteristici suportate, inclusiv versiunile Microsoft și Netscape ale HTML-ului, împreună cu gama lor de etichete de proprietar, cum ar fi <MARQUEE> și <LAYER>. Este practica industrială generală să se documenteze extensia sub forma unui DTD, care ar trebui să fie oferit de cel ce a introdus elementul.

Sunt disponibile mai multe DTD-uri, pentru diferite variațiuni HTML de la standard. Puteți găsi o mare colecție de DTD-uri și de alte resurse HTML la `www.webtechs.com/html/`.

Un DTD este exprimat ca o serie de declarații care identifică fiecare element, conținutul său posibil și contextul. Această identificare se face în primul rând prin folosirea a două unelte de bază: *entități* și *declarații*.

Entități

Dacă ați folosit vreodată un procesor de text pentru mult timp, atunci probabil v-ați dat seama cum să folosiți macrocomenzile. Macrocomenzile sunt acele mici scurtături folosite pentru a face diferite lucruri, cum ar fi introducerea numelui și a adresei la sfârșitul unei scrisori, scrierea unei fraze des folosite, ca de exemplu „Unde culoarea este exprimată prin trei valori hexazecimale", sau chiar scrierea unor paragrafe întregi.

Entitățile SGML sunt similare macrocomenzilor procesoarelor de text – ele sunt mici bucăți de text care se extind în șiruri sau instrucțiuni de marcare în cadrul DTD-ului.

În documentele HTML, entitățile sunt folosite să invoce caractere care fie lipsesc de pe tastatură (cum ar fi `é` pentru é), fie au un înțeles special și nu pot fi tastate direct în document (cum ar fi `<` și `>` pentru < și >, care încadrează de obicei informațiile etichetelor). În cadrul DTD-ului, entitățile au în special rolul de a face declarațiile mai concise și mai ușor de citit. Aceste entități folosite în DTD se numesc *entități parametru* spre deosebire de *entitățile generale* folosite în documentele HTML.

Pentru a începe, prima declarație în HTML DTD este o entitate parametru:

```
<!ENTITY % HTML.Version "http://www.w3.org/TR/WD-html40-970708/HTML4.dtd"
  -- Typical usage:

    <!DOCTYPE HTML SYSTEM "http://www.w3,org/TR/WD - html40-970708/HTML4.dtd">
    <html>
    ...
    </html>
  --
>
```

În primul rând observați sintaxa de delimitare, care este la fel pentru `ENTITY` și pentru `ELEMENT`. Începe cu paranteza ascuțită și un semn de exlamație și se termină cu o paranteză ascuțită. După cuvântul cheie `ENTITY` apare caracterul `%`, indicând faptul că entitatea respectivă este o entitate parametru și nu una generală.

Separat de un spațiu sau mai multe de caracterul procent este numele folosit la invocarea entității:

```
HTML.Version
```

Observați că numele conține un punct, sunt folosite setările secțiunii `NAMING` a declarației SGML, și că numele entităților sunt diferențiabile după tipul literelor (case-sensitive).

Ultima componentă obligatorie a declarației unei entități este șirul închis între ghilimele, care arată ce reprezintă această entitate și ce va rezulta când va fi invocată. Acesta este numit *șirul de date*:

```
"http://www.w3.org/TR/WD-html40-970708/HTML4.dtd"
```

Sintaxa necesară invocării entității în cadrul documentului SGML este:

```
%entityName
```

Deci, prima entitate din HTML DTD va fi văzută din nou sub următoarea formă:

```
%HTML.Version;
```

Ultima parte a declarației entității `HTML.Version` este comentariul, care aduce aminte de necesitatea începerii fiecărui document HTML ca document valid SGML, cu declarația `DOCTYPE`. Acest comentariu permite analizatoarelor SGML să realizeze imediat că structura și etichetele documentului ce urmează a fi procesat sunt descrise în DTD-ul identificat în URL. Desigur, browserele HTML se pot folosi și ele de această informație pentru a selecta nivelul suportului HTML necesar documentului, deși puține sunt acelea ce fac aceasta în realitate.

OBSERVAȚIE

Folosirea unui URL ca identificator DTD este ceva neobișnuit. Acest lucru este explicat prin faptul că în momentul scrierii acestei cărți, DTD pentru HTML încă evolua către o specificație finală. În general, pentru a face referire la surse de informații externe, documentele SGML folosesc niște *identificatori publici* de o formă specială.

Spre exemplu, DTD-ul HTML-ului 3.2 are o entitate `HTML.Version` care se extinde la `-//W3C//DTD HTML 3.2 Final//EN`, care este identificatorul public pentru DTD-ul HTML-ului 3.2. Când HTML 4 va ajunge la forma finală de implementare, identificatorul său public va fi `-//W3C//DTD HTML 4.0 Final//EN`. O formă scurtă, alternativă a identificatorului public este `-//W3C//DTD HTML 4.0//EN`.

Alt exemplu este șirul de identificare al unui set standard de caractere folosit pentru parametrul `BASESET` într-o declarație SGML. Orice DTD sau standard legat de acesta are un identificator public unic, care permite o referire din partea altor documente SGML. Aceste referiri se fac de obicei cu ajutorul entităților parametru.

Dacă șirul de date dintr-o entitate este precedat și de cuvântul cheie `PUBLIC`, atunci șirul nu este valoarea entității, ci un identificator public către o sursă externă de informații. Spre exemplu, DTD din HTML 4 este acompaniat de un set de entități generale pentru a accesa caracterele ISO Latin-1 – care reprezintă un standard izolat într-un document separat (`www.w3.org/pub/WWW/Markup/Cougar/ISOlat1.ent`), specificat de propriul său identificator public (`-/W3C//ENTITIES Latin1//EN/HTML`). Această informație este încorporată în DTD-ul din HTML sub forma următoare:

```
<!ENTITY & HTMLlat1 PUBLIC "-//W3C//ENTITIES Latin1//EN//HTML">
```

Acum, în loc să conțină un URL, șirul conține un identificator public al resursei externe, al cărui conținut înlocuiește fiecare apariție a entității `%HTMLlat1`. Regulile oficiale pentru construirea identificatorilor publici nu sunt date aici; mai multe informații precum și un catalog al identificatorilor pot fi găsite la `www.webtechs.com/html-tk/src/lib/catalog`.

De cealaltă parte se află *entitățile generale*. Ele sunt dclarate în DTD asemănător entităților parametru, dar cu câteva deosebiri importante:

- O entitate generală este folosită doar în documente care se conformează tipului de document. În acest caz este vorba despre un document HTML.
- O entitate generală nu conține caracterul `%` în declarația sa.
- O entitate generală este invocată folosind caracterul `&` (ampersand). De exemplu, `&lt` identifică caracterul „mai mic decât" (<). Deoarece acest lucru transformă ampersandul într-un caracter special, este necesară o entitate separată (`&amp`) pentru introducerea unui ampersand care să servească altor scopuri.
- O entitate generală conține de obicei cuvântul cheie `CDATA` (date de caracter) în declarația sa înainte de șirul de date. Cuvântul cheie indică faptul că șirul nu este interpretat ca date SGML. Cu alte cuvinte, orice instrucțiune de marcare ar putea conține acesta, va fi ignorată și tratată ca un text obișnuit.

Ca exemplu, priviți declarațiile acestor entități care definesc cum pot fi accesate patru caractere speciale:

```
<!ENTITY amp  CDATA "&#38;"   -- ampersand      -->
<!ENTITY gt   CDATA "&#62;"   -- greater than   -->
<!ENTITY lt   CDATA "&#60;"   -- less than      -->
<!ENTITY quot CDATA "&#34;"   -- double quote   -->
```

Acest exemplu prezintă un tip special de entitate, numit *referință de caracter*, care nu necesită nici o declarație. Dacă delimitatorul de deschidere al entității, `&`, este urmat de caracterul `#` și de un număr, acest număr este interpretat ca un cod al caracterului din setul de carctere al documentului definit în SGML de secțiunea `CHARSET`, iar întreaga entitate este înlocuită de caracterul corespunzător codului respectiv. Aceasta este una dintre cele două metode folosite pentru a accesa caractere ce se regăsesc dincolo de posibilitățile tastaturii; ea folosește entități cu denumirile mnemonice ale caracterelor, cum ar fi `&` sau `<`.

Poate vă întrebați cum pot entitățile din exemplul anterior să furnizeze caractere speciale din moment ce cuvânul cheie `CDATA` interzice instrucțiunilor SGML să aibă un efect în șirul de date. Răspunsul este că șirul este citit de două ori – prima dată când este interpretată declarația entității și a doua oară când însuși șirul de date este interpretat ca parte din document. Cuvântul cheie `CDATA` afectează doar prima citire. Rezultatul este că DTD-ul este protejat de caracterele speciale, iar documentul primește referințele extinse și caracterele care rezultă.

Elemente

Un document marcat printr-o aplicație SGML este de fapt o ierarhie de elemente imbricate. Un element marcat este de obicei închis într-o pereche de etichete, de început și de sfârșit. Declarația ELEMENT din SGML definește atât eticheta de început, cât și pe cea de sfârșit, dar nu și atributele acestora, și descrie conținutul elementului prin definirea modelului conținutului. Iată un exemplu de declarație a unui element:

```
<!ELEMENT P -O (%text)*>
```

În acest caz, P este numele elementului și reprezintă prescurtarea pentru *paragraf*. Cele două elemente care urmează numelui elementului sunt *indicatori de minimalizare*; ei specifică dacă este posibilă sau nu omiterea etichetelor de început sau de sfârșit ale acestui element. Primul indicator se referă la eticheta de început, iar cel de-al doilea la eticheta de sfârșit. Cele două posibilități pentru acest tip de notare sunt o cratimă sau un O. O cratimă înseamnă că este necesară o etichetă, iar O înseamnă că aceasta poate fi omisă. Deci, în exemplul anterior, un element P are obligatoriu o deschidere <P>, dar eticheta de sfârșit, </P>, este opțională.

Este posibil ca amândouă etichetele să fie opționale. Ca exemplu, priviți la declarația elementului HTML:

```
<!ELEMENT HTML OO (%html.content)>
```

Amândouă etichetele, atât cea de început, cât și cea de sfârșit, sunt opționale. Acest lucru înseamnă că puteți crea pagini Web corecte din punct de vedere tehnic fără să deschideți sau să închideți eticheta <HTML>? Desigur, este foarte corect. Pentru ca totul să fie clar, vă recomand totuși folosire lor, dar DTD-ul spune că nu sunteți obligat. Acest lucru este valabil și pentru etichetele de închidere ale celulelor de date din tabele, <TD>, ale paragrafelor, <P> precum și ale altor articole.

Acum priviți ultima componentă din declarațiile celor două elemente anterioare. Elementul P include

```
(%text)
```

iar elementul HTML are

```
(%html.content)
```

Cele două constituie specificația *modelului de conținut*. În acest caz este realizată cu entități parametru, iar pentru a vedea în ce se transformă prin extindere ar trebui să găsiți declarațiile ENTITY corespunzătoare în DTD. Modelul conținutului declară ce poate, ce nu poate și ce ar trebui să se afle în element.

Cel mai simplu tip al modelului de conținut este reprezentat de un singur cuvânt cheie din Tabelul 4.1.

Tabelul 4.1 Cuvinte cheie ale modelului de conținut.

Cuvânt cheie	*Descriere*
CDDATA	Reprezintă *datele caracterului* și face ca analizatorul SGML să suspende procesarea conținutului elementului. Etichetele sau entitățile aflate în element nu vor avea nici un efect asupra analizării și vor fi tratate precum orice alt caracter din text. Singura etichetă la care reacționează analizatorul SGML când trece peste conținutul CDATA este eticheta de sfârșit a elementului care a activat modul CDATA.

EMPTY	Arată că elementul nu are conținut. Întotdeauna acest cuvânt cheie este însoțit de permisiunea de a omite eticheta de sfârșit, după cum urmează: <!ELEMENT IMG -O EMPTY -- Embedded image -->
RCDATA	Înseamnă *date de caracter înlocuibile* și introduce un model de conținut care diferă de CDATA prin faptul că extinde toate entitățile generale și referințele la caracter, dar ignoră declarațiile de marcare. RCDATA nu este folosit în DTD-ul din HTML.
ANY	Permite accesul tuturor marcajelor și caracterelor de date în element. ANY nu este folosit în DTD-ul din HTML.

Câteodată este necesar să fiți mai explicit în definirea modelului de conținut al unui element. În acest caz, este folosit un *grup de modele de conținut*, aceasta fiind una dintre modalitățile de bază pentru definirea conținutului unui element. Această metodă este explicată în cele ce urmează.

Grupuri de modele de conținut

Cel mai simplu grup de modele este un nume de element închis între paranteze, care arată că elementul care este definit trebuie să aibă o parte a sa specificată în modelul de conținut și nimic altceva. Din nefericire, aceasta se întâmplă rar; mai des, un model conține două sau mai multe nume de element, după cum urmează:

```
<!ELEMENT DL - - (DT|DD)+>
```

Aceasta este declararea elementului pentru o listă de definiții. Dacă vă aduceți aminte din secțiunea anterioară, numele elementului este DL și sunt necesare ambele etichete, atât <DL>, cât și </DL>. Următoarea parte spune că în cadrul elementului DL, utilizatorul poate pune un element DT (termen), un element DD (definiție) sau pe amândouă. Cum vă închipuiți că se realizează acest lucru? Soluția este dată de separatorul dintre cele două elemente, numit *conector*. Acesta poate fi unul din trei caractere:

- O virgulă, în care caz elementele din conținut *trebuie* să apară în ordinea specificată. Acest separator este folosit foarte rar în DTD-ul din HTML, iar cea mai notabilă întrebuințare este în cazul elementului TABLE.
- O bară verticală reprezintă conectorul „sau". Arată că oricare element poate apărea, dar nu și amândouă. În cazul DL, este folosit acest conector pentru a permite cele două tipuri de conținut în listă.
- Un ampersand este conectorul „și". Acesta arată că toate elementele din listă trebuie să apară înaintea etichetei de închidere, dar spre deosebire de cazul elementelor despărțite prin virgulă, elementele pot apărea în orice ordine.

Semnul plus care urmează grupului în exemplu este un *indicator de apariții*; el definește de câte ori pot să apară elementele din grupul de conținut. Acest indicator poate fi unul dintre cele trei caractere:

- Un semn de întrebare înseamnă că elementul poate apărea o singură dată sau deloc, cum ar fi în cazul titlului unui tabel:

```
<!ELEMENT TABLE -- (CAPTION?,...
```

- Un plus înseamnă că elementele trebuie să apară măcar o dată, dar poate avea multe alte apariții. Ca exemplu, într-un formular este folosit un element SELECT pentru a crea o

listă autoderulantă, cu mai multe opțiuni. Prin urmare, poate include un număr nelimitat de opțiuni, dar trebuie să aibă cel puțin o apariție, după cum se vede:

```
<!ELEMENT SELECT - - (OPTION+)>
```

■ Un asterisc înseamnă că elementul poate apărea de orice număr de ori sau niciodată. Ca exemplu, o miniaplicație poate avea un parametru, mai mulți sau nici unul.

```
<!ELEMENT APPLET - - (PARAM|%inline)*>
```

Grupurile de modele pot fi imbricate și structurate prin folosirea unor paranteze adiționale, cu conectorii și indicatorii de apariții aplicați elementelor individuale sau grupurilor în întregime.

Pe lângă numele elementelor, este disponibil și cuvântul cheie `#PCDATA` pentru grupurile de modele. Acesta se referă la caracterele „obișnuite" ale documentului, fără nici un fel de etichete de marcare. Este folosit pentru a determina clar dacă este sau nu permisă existența textului simplu în document.

Folosirea cuvântului cheie `#PCDATA` împreună cu conectorii și cu indicatorii de apariții poate limita setul de elemente permise în interiorul altui element, fără a interzice apariția textului simplu în cadrul acestuia.

Ca exemplu, priviți cum este definită entitatea `%text` în cadrul unor entități subordonate:

```
<!ENTITY % font
 "TT | I | B | U | S | STRIKE | BIG | SMALL">

<!ENTITY % phrase   "EM | STRONG | DFN | CODE |
                     SAMP | KBD | VAR | CITE | ACRONYM">

<!ENTITY % special
    "A | IMG | APPLET | OBJECT | FONT | BASEFONT | BR | SCRIPT |
     MAP | Q | SUB | SUP | SPAN | BDO | IFRAME">

<!ENTITY % formctrl "INPUT | SELECT | TEXTAREA | LABEL | BUTTON">

<!-- %inline covers inline or "text-level" elements -->
<!ENTITY % inline "#PCDATA | %font | %phrase | %special | %formctrl">
```

În acest caz, fiecare entitate a fost construită cu o serie de posibile elemente, până la entitatea `inline`, care este definită ca date de tip caracter și/sau oricare dintre elementele `font`, `phrase`, `special` sau `formctrl` – pe înțelesul tuturor, „poate fi o bucată de text, pe lângă unul dintre celelalte elemente listate".

După ce entitățile sunt definite, ele pot fi folosite să creeze alte modele de conținut, ca în exemplul următor:

```
<!ELEMENT (%font|%phrase) - - (%text)*>
```

Această declarație înseamnă că un element `font` sau `phrase` are în mod obligatoriu o etichetă de început și una de sfârșit și poate include orice număr de caractere de text.

Altă caracteristică interesantă a grupurilor de conținut este folosirea caracterelor pentru adăugarea sau extragerea unor grupuri de modele. De exemplu, elementului `FORM` i se permite să conțină orice poate apărea într-un element la nivel de bloc, cu excepția altui element `FORM`. Deoarece aceasta înseamnă că formularele nu pot fi conținute, este folosită următoarea operație de extragere:

```
<!ELEMENT FORM - - %block.content -(FORM)>
```

Acum aveți elementele la îndemână, dar, după cum știți, multe etichete conțin și atribute. Acesta este subiectul următoarei secțiuni.

Atribute pentru elemente

Un element nu este descris pe deplin de modelul de conținut și de numele său. Multe elemente sunt însoțite de *atribute*, care furnizează informații adiționale despre redarea elementului. Atributele fiecărui element sunt declarate în DTD cu instrucțiunea `ATTLIST`.

Iată lista de atribute pentru elementul linie orizontală:

```
<!ATTLIST HR
  %coreattrs;                       -- id, class, style, title --
  %events;
  align (left|right|center) #IMPLIED
  noshade    (noshade)      #IMPLIED
  size       %Pixels        #IMPLIED
  width      %Length        #IMPLIED
  >
```

Primele două atribute sunt numele entităților care se extind către un set generic de atribute disponibile pentru practic toate elementele. Atributele principale (`coreattrs`) sunt folosite în primul rând pentru implementarea foilor de stil, iar atributele de eveniment (`events`) acoperă evenimentele standard ale documentului.

Urmează atributele care aparțin doar elementului `HR`. Fiecare intrare începe cu numele atributului, cum ar fi `align`, apoi continuă cu tipul de valoare pe care o poate primi atributul și cu valoarea prestabilită. Continuând cu prima intrare, cele trei posibilități sunt `left`, `right` sau `center`.

Unele atribute primesc o valoare prestabilită care depinde de aspectul lor. Acest caz este întâlnit în elementul `HR` cu atributul `noshade`:

```
noshade (noshade) #IMPLIED
```

Efectul acestei instrucțiuni este să furnizeze două metode posibile pentru specificarea atributului când acesta este folosit în marcare: `<HR noshade=noshade>` sau `<HR noshade>`. La fel ca la grupurile de conținut pentru elemente, singura valoare dintre parantezele definiției atributului indică numai o singură valoare ca fiind posibilă. Dacă atributului nu i se atribuie o valoare explicită, este folosită valoarea necesară.

Pe lângă valorile denumite, aflate între paranteze, valoarea mai poate fi unul dintre următoarele cuvintele cheie:

■ **`CDATA`:** Acest cuvânt cheie înseamnă că valoarea atributului poate fi orice șir de caractere și ar trebui să fie ignorat de analizator. `CDATA` este folosit în situații în care ar fi imposibilă impunerea unor limitări mai stricte asupra valorii unui atribut, cum ar fi în cazul unei diviziuni sau al unui element de formular.
■ **`NAME`:** Acest cuvânt cheie indică valoarea unui atribut ca fiind un nume în conformitate cu regulile de denumire definite în declarația SGML.
■ **`NMTOKEN`:** Acest cuvânt cheie este similar cu `NAME`, cu excepția că nu este nici o cerință de a începe numele cu caracterul de început de nume (de obicei, un caracter latin).
■ **`NUMBER`:** Acest cuvânt cheie permite parametrului să ia valori numerice, cum ar fi numărul de început al unei liste ordonate:

```
<!ATTLIST OL -- ordered lists --
...
start  NUMBER   #IMPLIED  -- starting sequence number --
...
>
```

■ **ID:** Ultimul cuvânt cheie arată că valoarea atributului este un *identificator* care satisface două cerințe. În primul rând este un nume SGML valabil, iar în al doilea rând este unic în document (nu este atribuit nici unui alt atribut). Această valoare este specifică atributului ID pentru foi de stil.

Ultima parte a definirii atributului precizează valoarea prestabilită. În majoritatea exemplelor anterioare, a fost folosit cuvântul cheie #IMPLIED; totuși, poate fi în locul său o valoare anume. Luați în considerare următorul pasaj din elementul FORM:

```
<!ELEMENT FORM - - %block - (FORM)>
<!ATTLIST FORM
...
 method     (GET|POST) GET     -- HTTP method used to submit the form --
...
>
```

Atributul method este forțat să ia valoarea prestabilită GET dacă nu este identificat explicit, deși poate lua atât valoarea GET, cât și POST.

Majoritatea atributelor au unul dintre următoarele trei cuvinte pe post de valoare prestabilită:

■ **#FIXED:** Acest cuvânt cheie trebuie să fie înaintea valorii prestabilite și este folosit pentru a arăta că valoarea nu poate fi schimbată de utilizator. Este folosit doar o singură dată în DTD-ul din HTML:

```
<!ATTLIST HTML
     VERSION CDATA #FIXED "%HTML.version;"
...
>
```

■ **#IMPLIED:** Acest cuvânt cheie arată atributul ca fiind opțional. Browserul poate folosi opțional valoarea sa prestabilită, deși nu este menționată o valoare prestabilită în DTD.
■ **#REQUIRED:** Acest cuvânt cheie arată că atributul trebuie inclus. Ca exemplu, priviți această linie din lista atributului FORM:

```
action    %URL    #REQUIRED -- server-side form handler --
```

Această linie arată că atributul action trebuie să aibă un URL specificat ca un handler de formular.

Caracteristici nerecomandate

DTD HTML include mai multe *caracteristici nerecomandate*, care sunt elemente și atribute ce nu mai sunt indicate a fi folosite, dar sunt incluse în DTD pentru a asigura compatibilitatea cu versiunile anterioare.

SUGESTIE

Evitați utilizarea caracteristicilor nerecomandate în documentele dumneavoastră. Deși ele pot fi în continuare acceptate de multe browsere, fiind nerecomandate înseamnă că sunt pe cale de dispariție. Există șansa de a exista o variantă mai bună prin care să obțineți rezultatul dorit, cum este cazul listei neordonate pentru conținutul unui director.

Elementele nerecomandate sunt uneori procesate diferit față de restul documentului. Pentru a face aceasta, SGML oferă o metodă de identificare a *secțiunilor marcate*, ceea ce face posibilă izolarea oricărei instrucțiuni sau declarații de marcare pentru a controla procesarea ei:

```
<![ %HTML.Deprecated [
    <!ENTITY % preformatted "PRE | XMP | LISTING">
]]>
```


Entitatea %HTML.Deprecated se extinde într-un cuvânt cheie special, care comunică analizatorului ce să facă în ceea ce privește conținutul acelei secțiuni. Dacă cuvântul este IGNORE, atunci secțiunea marcată este ignorată în întregime. Dacă acesta este INCLUDE, atunci secțiunea marcată este procesată exact ca restul DTD. Deci, pentru a obține o versiune a DTD-ului care să conțină doar elementele recomandate, entitatea HTML.Deprecated ar trebui schimbată din

```
<!ENTITY % HTML.Deprecated "INCLUDE">
```

în

```
<!ENTITY % HTML.Deprecated "IGNORE">
```

Rezumat

Am acoperit mult teren în acest capitol, dar acum sper că aveți idee despre ce este un DTD și cum trebuie acesta citit. Am folosit standardul W3C în exemplele din acest capitol, dar dacă treceți în revistă oricare din DTD-urile altor variante de HTML (cum ar fi DTD-urile disponibile la www.webtechs.com/html), veți observa că aceleași standarde și sintaxe se aplică la toate.

Pe scurt, DTD-ul din HTML este o aplicație SGML ruptă în două părți majore: declarația și DTD-ul. Declarația are grijă de treburile casei, cum ar fi ce set de caractere să fie folosit și pe ce versiune de SGML se bazează restul documentului. DTD-ul include definiții pentru toate elementele HTML și atributele acestora, folosind o sintaxă care se aseamănă foarte tare cu HTML-ul. Caracterele speciale și separatorii ajută la specificarea conținutului și structurii folosirii elementului.

Este nevoie să citiți DTD-ul în întregime pentru a crea documente HTML? Nu, tot ce vă trebuie este o carte bună ca aceasta. Totuși, dacă încercați să creați pagini în concordanță cu anumite tipuri de HTML sau vreți să vedeți dacă au apărut schimbări importante într-un nou standard HTML, puteți citi direct DTD-ul ca să vedeți despre ce este vorba.

Capitolul 5
Acum, să începem

De Rick Darnell

În acest capitol

- **Problematica de bază a machetei paginii 48**
- **Pagina 1 – un șablon 56**
- **O pagină cu o listă 63**

Restul acestei cărți urmează să analizeze amănunțit fiecare element și atribut din spe ficația HTML 4.0, dar acum să privim pe scurt la ceea ce conține HTML-ul și cum construiește o pagină Web. În cazul în care nu ați mai lucrat cu HTML-ul înainte sau do căutați niște idei noi, atunci acest capitol este pentru dumneavoastră.

Acest capitol se concentrează pe construirea câtorva pagini HTML care să încorpore majoritatea caracteristicilor standard HTML. De asemenea, am să vă ofer o introducere designul elementar cu HTML, care este continuat în Partea aVII-a („Designul efectiv unei pagini Web"), care discută mult mai detaliat designul unei pagini Web.

Problematica de bază a machetei paginii

Înainte de a începe să creați o pagină, trebuie să răspundeți la două întrebări importante

- Ce vreți să spuneți?
- Cui vreți să-i spuneți?

Deși sunt două întrebări diferite, sunt greu de despărțit. Ceea ce doriți să spuneți depind de cui vreți să-i spuneți. Pe măsură ce veți încerca să răspundeți acestor două întrebări probabil că vor apărea și altele în timpul planificării paginii dumneavoastră de Wel Următoarele întrebări vă vor ajuta să vă stabiliți și mai precis obiectivele:

- Paginile dumneavoastră sunt adresate potențialilor clienți?
- Doriți să furnizați informații unice despre un anumit subiect?
- Este această pagină o metodă de a vă menține angajații și restul personalului la curent cu ultimele noutăți și proiecte din cadrul companiei?

Dacă nu aveți o audiență sau un mesaj, există șansa să nu spuneți nimic nimănui Acest sfat este important și pentru Capitolul 15, „Crearea și utilizarea formularelo HTML". Formularul este conform funcției sale, iar neajunsurile unuia se manifestă ș în celălalt.

După ce stabiliți care este mesajul dumneavoastră de bază și cum va fi organizat pentru a perceput de audiență, puteți începe să plasați componentele individuale ale mesajului î cadrul de lucru. Dacă am asemăna situl Web cu un clopot, atunci paginile ar fi limbil care-l izbesc pentru a produce sunetul.

Înainte de a începe o nouă pagină, amintiți-vă că designul și macheta paginii reprezint modul în care mesajul este comunicat utilizatorului. Un design și o machetă bună nu sun scopuri în sine. Dacă vă uitați la multe pagini pe Web, veți vedea toată gama – de la pagin parcă nefinisate, în care nu se regăsește nici o intenție de înfrumusețare a aspectului, pân

la pagini care nu au alt mesaj decât să arate cât de artistic poate fi un design. Din moment ce începeți să vă jucați cu designul, amintiți-vă regula de bază a machetei: Delăsarea ucide mesajul.

Este o idee bună să răsfoiți reviste, cărți și broșuri înainte de a începe. Designul paginii Web are la bază designul paginilor tipărite, deci o pagină tipărită este un loc bun să începeți să căutați idei.

Deși numărul opțiunilor și capacităților de formatare este în creștere pentru paginile Web, aceste caracteristici sunt încă limitate în comparație cu cele disponibile pentru verișoarele lor tipărite. Cu un pic de creativitate puteți totuși să faceți ca pagina dumneavoastră să iasă în evidență dintre paginile obișnuite.

CÂTEVA CUVINTE DESPRE SPAȚIU

Spațiul este o parte importantă a fiecărei pagini. Te lasă să-ți mai tragi sufletul în încâlceala unei mulțimi de informații. Folosirea eficientă a spațiului poate ajuta unei pagini în aceeași măsură în care folosirea ineficientă poate fi în detrimentul acesteia.

Dacă o pagină conține prea mult spațiu, utilizatorul va pierde mult timp defilând pentru a ajunge la informația următoare. Dacă spațiul este prea mic, totul se va contopi, ca într-o ciorbă foarte groasă. O altă opinie vă poate ajuta să apreciați efectul total. Cereți prietenilor și colegilor de serviciu să-și spună părerea despre lucrarea dumneavoastră. Acestea sunt unele dintre cele mai bune sfaturi pe care le puteți obține pe gratis.

Următoarele cinci machete folosesc fontul prestabilit pe care îl furnizează browserul și aceeași imagine și mărime a titlului principal. În ciuda acestor limitări, modalitățile diferite de plasare a titlului, textului, imaginilor și spațiilor albe rezultă în aspecte foarte variate pentru fiecare machetă în parte. În funcție de cum folosiți și așezați cele patru elemente ale paginii, fiecare machetă se poate personaliza, fapt ce îl puteți folosi pentru a vă sublinia mesajul.

O pagină convențională

Primul stil de pagină este unul ce poate fi văzut foarte des pe Web. Este și cel mai ușor de construit, și nici nu necesită cunoștințe solide în ceea ce privește designul sau HTML-ul. O pagină convențională nu se folosește de nici o caracteristică HTML de formatare, dar este compatibilă cu toate browserele și platformele. Ea conține un titlu urmat de corpul textului și poate o imagine la sfârșit sau în altă parte. (Vezi Listingul 5.1 și Figura 5.1).

Listingul 5.1. Macheta unei pagini convenționale.

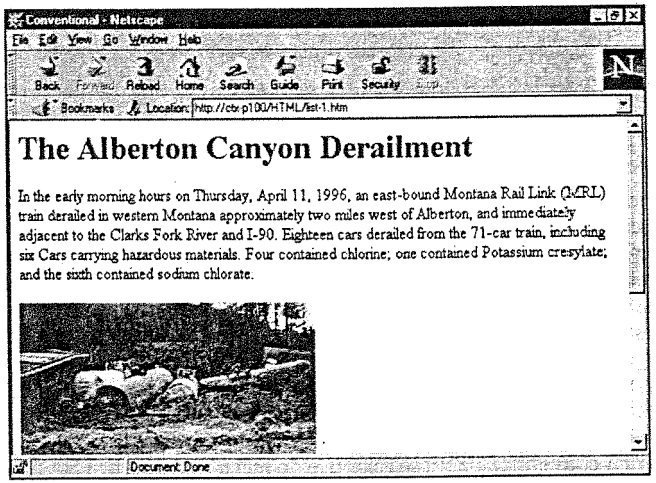
```
<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 4.0//EN">
<HTML>
<HEAD>
  <TITLE>Conventional</TITLE>
</HEAD>
<BODY>
<H1>The Alberton Canyon Derailment</H1>
<P>In the early morning hours on Thursday, April 11, 1996, an east-bound Montana
Rail Link (MRL) train derailed in western Montana approximately two miles west of
Alberton, and immediately adjacent to the Clarks Fork River and I-90. Eighteen
cars derailed from the 71-car train, including six Cars carrying hazardous
materials. Four contained chlorine; one contained Potassium cresylate; and the
sixth contained sodium chlorate.</P>
<P><IMG SRC="/mrfd/Alberton/albert8.jpg" WIDTH="300" HEIGHT="197"></P>
```

(continuare)

Listingul 5.1. Continuare

```
<P>To compound the problem further, another MRL train derailed near Noxon,Mont.,
spilling 1,000 gallons of diesel fuel and overturning another chlorine car. The
additional confusion caused by the diversion and rerouting of resources slowed
the initial response of personnel and equipment.</P>
<P>One of the chlorine cars and the potassium cresylate car were breached,
resulting in the release of approximately 100,000 lbs. of deadly chlorine gas.
The cloud drifted over the highway and through the town of Alberton forcing the
evacuation of more than 1,000 people in a 100 square-mile area. Half of these
were kept out of their homes for 17 days. An estimated 350 people were treated
for health and respiratory problems in Missoula hospitals, 35 miles to the east.
A lone fatality occurred when a transient riding in a box car walked into the
cloud after surviving the crash.</P>
</BODY>
</HTML>
```

FIGURA 5.1
O pagină convențională: titlu la început, text compact la mijloc și ilustrație la sfârșit.



Deși nu vă doriți ca toate paginile din situl dumneavoastră să arate ca aceasta, este c metodă rapidă și eficientă să alcătuiți o pagină pentru a furniza informația, până când puteți face una mai bună. Este de asemenea o soluție bună dacă intenționați să alcătuiți c pagină pentru browsere mai vechi, care s-ar putea să nu suporte caracteristicile avansate ale HTML-ului 4.0.

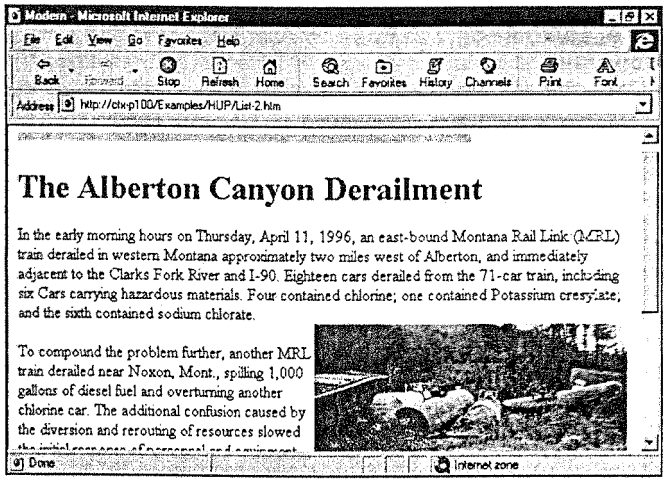
O machetă modernă

O machetă modernă este asemănătoare cu cea convențională, a textului compact, dar în ea se folosesc linii orizontale pentru a marca începutul și sfârșitul paginii și imagini care se extind în text. (Vezi Listingul 5.2 și Figura 5.2). O machetă cu adevărat modernă conține spații mai mari între rândurile textului (veți vedea cum se face acest lucru în Capitolul 18 „Utilizarea foilor de stil în cascadă"). Principala caracteristică a unei machete moderne este folosirea alinierii la dreapta și la stânga a imaginii, permițând textului să curgă în jurul acesteia în loc să stea compact.

Listingul 5.2. O machetă modernă

```
<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 4.0//EN">
<HTML>
<HEAD>
  <TITLE>Modern</TITLE>
</HEAD>
<BODY>
<HR NOSHADE STYLE="width:75%; height:8px; text-align:left">
<H1>The Alberton Canyon Derailment</H1>
<P>In the early morning hours on Thursday, April 11, 1996, an east-bound Montana
Rail Link (MRL) train derailed in western Montana approximately two miles west of
Alberton, and immediately adjacent to the Clarks Fork River and I-90. Eighteen
cars derailed from the 71-car train, including six Cars carrying hazardous
materials. Four contained chlorine; one contained Potassium cresylate; and the
sixth contained sodium chlorate.
<IMG SRC="/mrfd/Alberton/albert8.jpg" ALIGN="RIGHT"
➥STYLE="width:300px; height:197px"></P>
<P>To compound the problem further, another MRL train derailed near Noxon,Mont.,
spilling 1,000 gallons of diesel fuel and overturning another chlorine car. The
additional confusion caused by the diversion and rerouting of resources slowed
the initial response of personnel and equipment.</P>
<P>One of the chlorine cars and the potassium cresylate car were breached,
resulting in the release of approximately 100,000 lbs. of deadly chlorine gas.
The cloud drifted over the highway and through the town of Alberton forcing the
evacuation of more than 1,000 people in a 100 square-mile area. Half of these
were kept out of their homes for 17 days. An estimated 350 people were treated
for health and respiratory problems in Missoula hospitals, 35 miles to the east.
A lone fatality occurred when a transient riding in a box car walked into the
cloud after surviving the crash.</P>
<HR noshade style="width:75%; height:8px; text-align:right">
</BODY>
</HTML>
```

FIGURA 5.2.
O pagină mai modernă, care folosește linii separatoare orizontale și imagini integrate în text.



Puteți foarte ușor să schimbați o machetă convențională într-una modernă prin adăugarea parametrilor `left` și `right` atributului `align` al unei imagini. Acest lucru îmbunătățește vizibil aspectul paginii fără să depuneți un efort prea mare.

O machetă clasică

Adăugarea de coloane unei pagini are două efecte. În primul rând, scurtează lungimea rândului, făcând textul mai ușor de citit de către utilizator. Acest lucru este foarte important în special pentru o pagină Web, deoarece majoritatea utilizatorilor o privesc pe un ecran, ceea ce cauzează un stres mai mare asupra ochilor decât în cazul citirii unei pagini tipărite.

În al doilea rând, coloanele economisesc spațiul în documentele mari. Deși, aparent, coloanele ocupă mai mult loc, ele reduc lungimea totală a documentului, aceasta fiind și cauza pentru care ziarele folosesc paginația pe coloane multiple.

O machetă clasică este una dintre cele mai simple întrebuințări a coloanelor într-o pagină Web. (Vezi Listingul 5.3 și Figura 5.3). Deoarece HTML nu are un suport direct pentru coloane, tabelele se achită de această sarcină destul de bine, obținându-se același efect.

Listingul 5.3. O machetă clasică care folosește tabele pentru a crea coloane.

```
<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 4.0//EN">
<HTML>
<HEAD>
 <TITLE>Classic</TITLE>
</HEAD>
<STYLE type="text/css">
<!--
.center {text-align:center}
-->
</STYLE>
<BODY>
<H1 CLASS="center">The Alberton Canyon Derailment</H1>
<HR STYLE="width:75%; height:8px" CLASS="center" NOSHADE>
<TABLE WIDTH="100%" BORDER="0" CELLSPACING="15">
 <TR VALIGN="TOP">
  <TD WIDTH="50%">
   <P>In the early morning hours on Thursday, April 11, 1996, an east-bound
   Montana Rail Link (MRL) train derailed in western Montana approximately two
   miles west of Alberton, and immediately adjacent to the Clarks Fork River and
   I-90. Eighteen cars derailed from the 71-car train, including six Cars
   carrying hazardous materials. Four contained chlorine; one contained Potassium
   cresylate; and the sixth contained sodium chlorate.
   <P><IMG SRC="/mrfd/Alberton/albert8.jpg" STYLE="width:300px; height:197px"></P>
  <TD WIDTH="50%">
   <P>To compound the problem further, another MRL train derailed near
   Noxon,Mont., spilling 1,000 gallons of diesel fuel and overturning another
   chlorine car. The additional confusion caused by the diversion and rerouting
   of resources slowed the initial response of personnel and equipment.</P>
   <P>One of the chlorine cars and the potassium cresylate car were breached,
   resulting in the release of approximately 100,000 lbs. of deadly chlorine gas.
   The cloud drifted over the highway and through the town of Alberton forcing
   the evacuation of more than 1,000 people in a 100 square-mile area. Half of
   these were kept out of their homes for 17 days. An estimated 350 people were
   treated for health and respiratory problems in Missoula hospitals, 35 miles to
   the east. A lone fatality occurred when a transient riding in a box car walked
   into the cloud after surviving the crash.</P>
```



```
</TABLE>
</BODY>
</HTML>
```

Această machetă, arătată în Figura 5.3, se folosește de tabele pentru a crea o pagină Web. Titlul este plasat în afara tabelului și este centrat pentru a ieși în evidență.

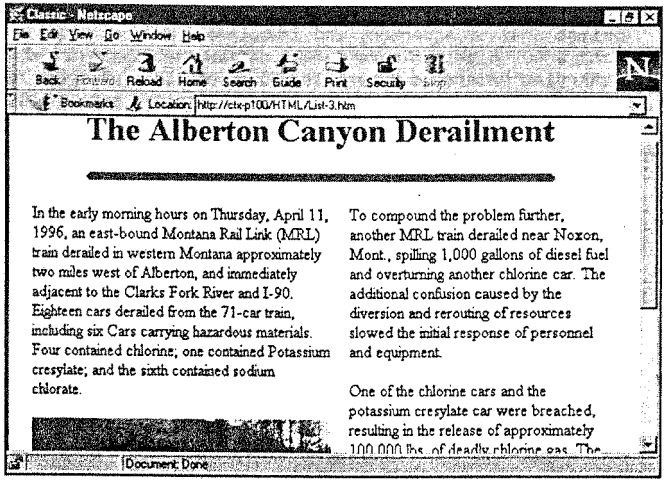


FIGURA 5.3. *Această machetă clasică este un format cu două coloane care are titlul centrat și un set de imagini în text.*

Cele două coloane trebuie să fie la fel de late, una singură necesitând atributul `width`. Totuși, amândouă etichetele `<TD>` includ specificația `width`, ceea ce vă ajută să evitați ambiguitatea în cazul îmbunătățirii sau verificării ulterioare a paginii.

O pagină pentru informații tehnice

Macheta pentru documente tehnice a apărut în urma necesității unor caracteristici practice. La început, multe pagini semănau mai mult cu un design convențional. Aceastei machete îi lipsea spațiul necesar unor adnotări sau comentarii despre conținut. Prin trecerea la un format pe trei coloane și lăsarea uneia dintre coloanele exterioare liberă pentru diagrame și ilustrații, pagina a câștigat un plus de spații albe.

Designul rezultat este foarte simplu, iar spațiile albe rezultate dau paginii o înfățișare curată și puternică. (Vezi Listingul 5.4 și Figura 5.4).

Listingul 5.4. Macheta unei pagini pentru informații tehnice.

```
<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 4.0//EN">
<HTML>
<HEAD>
  <TITLE>Tehnical</TITLE>
</HEAD>
<BODY>
<TABLE BORDER="0" WIDTH="100%" CELLSPACING="15">
  <TR>
    <TD>
    <TD COLSPAN="2" WIDTH="33%">
      <H1>The Alberton Canyon Derailment</H1>
```

(continuare)

Listingul 5.4. Continuare

```
  <TR VALIGN="TOP">
    <TD VALIGN="MIDDLE">
      <P><IMG SRC="/mrfd/Alberton/albert8.jpg" STYLE="width:225; height:148"></P>
    <TD WIDTH="33%">
      <P>In the early morning hours on Thursday, April 11, 1996, an east-bound
       Montana Rail Link (MRL) train derailed in western Montana approximately two
       miles west of Alberton, and immediately adjacent to the Clarks Fork River
       and I-90. Eighteen cars derailed from the 71-car train, including six Cars
       carrying hazardous materials. Four contained chlorine; one contained
       Potassium cresylate; and the sixth contained sodium chlorate.</P>
      <P>To compound the problem further, another MRL train derailed near
      Noxon,Mont., spilling 1,000 gallons of diesel fuel and overturning another
      chlorine car. The additional confusion caused by the diversion and rerouting of
      resources slowed the initial response of personnel and equipment.</P>
    <TD WIDTH="33%">
      <P>One of the chlorine cars and the potassium cresylate car were breached,
       resulting in the release of approximately 100,000 lbs. of deadly chlorine
       gas. The cloud drifted over the highway and through the town of Alberton
       forcing the evacuation of more than 1,000 people in a 100 square-mile area.
       Half of these were kept out of their homes for 17 days. An estimated 350
       people were treated for health and respiratory problems in Missoula
       hospitals, 35 miles to the east. A lone fatality occurred when a transient
       riding in a box car walked into the cloud after surviving the crash.</P>
       </TABLE>
</BODY>
</HTML>
```

Designul paginii tehnice, arătat în Figura 5.4, este asemănător clasicului format pe două coloane, cu excepția că o a treia coloană a fost introdusă pentru a conține imagini, comentarii, diagrame și altele.

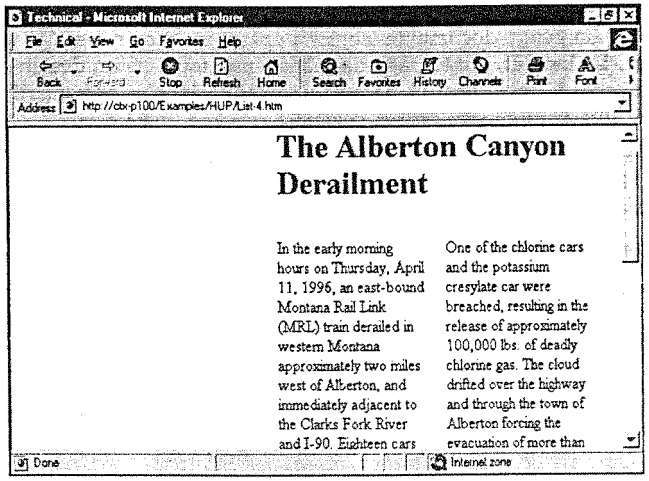


FIGURA 5.4. *Acest design combină unele caracteristici ale machetelor convenționale și moderne, dar folosește mai mult spațiu alb.*

O pagină formală

Paginile formale nu sunt neapărat cele mai eficiente designuri din lume, dar au un anumit mod care este potrivit pentru anunțuri, cum ar fi numirea unui nou șef la departamentul local de pompieri sau dotarea departamentului respectiv cu o nouă mașină.

Designul este creat prin folosirea unui tabel cu trei coloane. Lățimea fiecărei coloane exterioare este de 25 la sută din lățimea tabelului, lăsând jumătate din lățimea tabelului coloanei centrale și textului (după cum puteți vedea în Listingul 5.5 și Figura 5.5). Fiecare element conținut este plasat în propria sa celulă – întâi titlul, apoi imaginea și la urmă, textul.

Listingul 5.5. O machetă a unei pagini formale.

```
<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 4.0//EN">
<HTML>
<HEAD>
  <TITLE>Formal</TITLE>
</HEAD>
<STYLE type="text/css">
.center {text-align:center}
.initial {font-size:+3}
</STYLE>
<BODY>
<TABLE BORDER="0" WIDTH="100%" CELLSPACING="3">
  <TR>
    <TD WIDTH="25%">
    <TD>
      <H1 class="center">The Alberton <BR>
       Canyon Derailment</H1>
  <TD WIDTH="25%">
<TR>
  <TD>
  <TD>
    <P class="center">
      <IMG SRC="/Mrfd/Alberton/albert8.jpg" STYLE="width:300px; height:197px"
      ➥VSPACE="24">
    </P>
  <TD>
<TR>
  <TD WIDTH="25%">
  <TD>
    <P><SPAN class="initial">I</SPAN>n the early morning hours on Thursday, April
     ➥11, 1996, an east-bound Montana Rail Link (MRL) train derailed in western
     Montana approximately two miles west of Alberton, and immediately adjacent
     to the Clarks Fork River and I-90. Eighteen cars derailed from the 71-car
     train, including six Cars carrying hazardous materials. Four contained
     chlorine; one contained Potassium cresylate; and the sixth contained sodium
     chlorate.</P>
    <P>To compound the problem further, another MRL train derailed near Noxon,Mont.,
     spilling 1,000 gallons of diesel fuel and overturning another chlorine car. The
     additional confusion caused by the diversion and rerouting of resources slowed
     the initial response of personnel and equipment.</P>
    <P>One of the chlorine cars and the potassium cresylate car were breached,
     resulting in the release of approximately 100,000 lbs. of deadly chlorine
```

(continuare)

Listingul 5.5. Continuare

```
      gas. The cloud drifted over the highway and through the town of Alberton
      forcing the evacuation of more than 1,000 people in a 100 square-mile area.
      Half of these were kept out of their homes for 17 days. An estimated 350
      people were treated for health and respiratory problems in Missoula
      hospitals, 35 miles to the east. A lone fatality occurred when a transient
      riding in a box car walked into the cloud after surviving the crash.</P
        <TD WIDTH="25%"
</TABLE
</BODY
</HTML
```

Într-un design formal, trei coloane creează spațiu la stânga și la dreapta conținutului și controlează antetul și spațiul inițial rezervat imaginii. Macheta formală este în mod special potrivită acelor momente când contează doar demnitatea. În Figura 5.5, literele capitale mari, plasamentul central al imaginii și al textului, precum și mult spațiu liber, dau acestei pagini grație și simplitate.

FIGURA 5.5.
Un exemplu de design formal.

Ca o caracteristică adăugată, puteți mări prima literă din text prin folosirea unei foi de stil, care vă permite în plus formatări ale textului și folosirea altor fonturi.

Pagina 1 – un șablon

Având aceste informații de bază despre machete, este timpul să vă suflecați mânecile și să vă apucați de treabă. Folosind situl Departamentului de Pompieri al Districtului Missoula ca exemplu, veți începe cu un șablon care va deveni baza celorlalte pagini ale dumneavoastră.

OBSERVAȚIE

Un șablon HTML servește drept ghid pentru alte pagini HTML, după cum este model pentru decuparea sau desenarea contururilor. Șabloanele sunt folositoare pentru menținerea aceluiași design și imagine pe mai multe pagini, fără muncă la întâmplare, prin furnizarea unei mari părți a structurii de bază și a formatării înainte să începeți.

Departamentul de pompieri își folosește situl pentru a oferi publicului o varietate de informații, inclusiv sfaturi și evenimente legate de anumite date calendaristice. Ei folosesc,

de asemenea, situl pentru a face cunoscute comunității diferitele lor atribuții și responsabilități. Paginile Web trebuie să aibă un aspect profesional și totuși să ofere informațiile într-o manieră prietenoasă.

Pentru realizarea acestui scop, au ales un design care este o îmbinare între machetele clasice și cele tehnice. Include spații albe mari pe partea stângă, întrerupte doar de titluri agățate și de imaginea corespunzătoare, în combinație cu o manșetă scrisă cu aldine deasupra.

Construcția paginii începe de la bază – cu structura documentului (după cum se vede în Listingul 5.6).

Listingul 5.6. Începutul șablonului pentru un set de pagini Web.

```
<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 4.0//EN">
<HTML>
<HEAD>
<TITLE>MRFD - a title</TITLE>
</HEAD>
<STYLE type="text/css">
<!--
BODY {background:white}
-->
</STYLE>
<BODY>
</BODY>
</HTML>
```

Structura nu creează o pagină, dar chiar și un castel are nevoie de o fundație înainte de a flutura steagurile în turnuri. Pe scurt, lucrul la o pagină Web ar trebui să înceapă cu începutul. Pe lângă structură, veți specifica o culoare prestabilită pentru fundal cu ajutorul unei foi de stil. De asemenea, este adăugat marcajul de rezervare `<TITLE>`, care conține formatul pentru titlurile paginii Web, pentru a nu uita să adăugați un titlu în paginile de mai târziu.

SUGESTIE

Este o idee bună să începeți o pagină cu un identificator al sitului dumneavoastră. Acest identificator va face mai ușoară localizarea pentru utilizatori dacă ei vor include un semn pentru pagină pe browserul lor de acasă.

Acum, că primul detaliu a fost explicat, este timpul să ne apucăm de formatarea începutului documentului. Pentru a obține un aspect unitar, plasați aceeași manșetă la începutul fiecărei pagini, prin folosirea unui tabel simplu cu text și imagine. (Vedeți Listingul 5.7 și Figura 5.6). Această tehnică este descrisă detaliat în Capitolul 10, „Macheta și prezentarea tabelelor".

Listingul 5.7. Prima parte a paginii include un tabel pe două coloane cu un singur rând.

```
<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 4.0//EN">
<HTML>
<HEAD>
<TITLE>MRFD - a title</TITLE>
</HEAD>
<STYLE type="text/css">
<!--
BODY {background:white}
TABLE.banner {background:black}
```

(continuare)

Listingul 5.7. Continuare

```
H1.banner {color:white}
-->
</STYLE>
<BODY>
<TABLE class="banner" border="0" cellspacing="15" width="100%">
 <TR>
  <TD align="RIGHT" valign="BOTTOM" width="75%">
   <H1 class="banner">...Subject Title...</H1>
  <TD align="LEFT" valign="BOTTOM">
   <IMG src="/mrfd/images/fire2.jpg" HEIGHT="158" WIDTH="149">
</TABLE>
</BODY>
</HTML>
```

Ar trebui să luați în considerare mai multe lucruri la deschiderea etichetei de tabel. Î primul rând, chenarele sunt invizibile (`borders="0"`), ceea ce face ca numai formatare propriu-zisă să fie vizibilă utilizatorului; metoda în care a fost aceasta realizată rămân ascunsă. Apoi, atributul `cellspacing` folosește o cifră mai mare decât în mod norma pentru a umple conținutul celor două celule de date. Tabelul va ocupa întotdeauna sută l sută din lățimea disponibilă a tabelului, iar când este clasificat ca fiind `banner`, va ave fundalul negru (`background:black`).

Conținutul primei celule este forțat către marginile de jos și din dreapta și va ocupa 75 l sută din lățimea disponibilă a tabelului. Apoi titlul paginii este adăugat cu alb pentru contrasta cu fundalul negru. Deoarece este titlul principal, veți folosi eticheta `<H1>`.

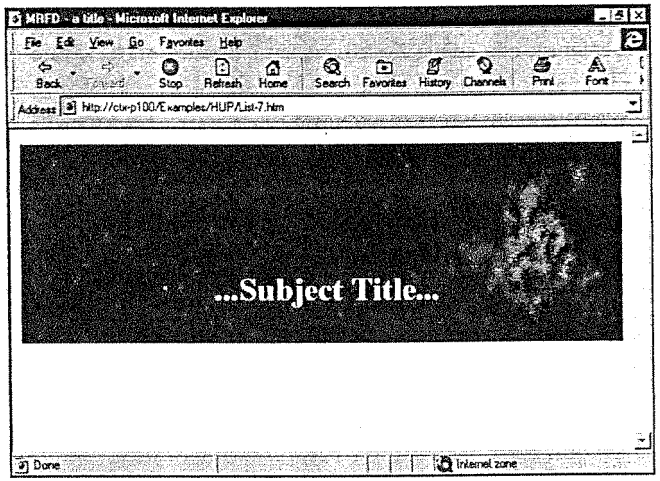


FIGURA 5.6.
Șablonul, cum apare până acum, cu titlul documentului și antetul pentru partea superioară a documentului.

Conținutul celei de-a doua celule este aliniat la marginile din stânga și de jos. Atribut `width` nu este folosit deoarece browserul va atribui automat acesteia restul de 25 la sută d tabel. Conținutul său este o image JPEG cu specificațiile `height` și `width` (înălțime și lățime (Pentru mai multe informații despre aceste atribute, citiți Capitolul 11, „Adăugarea imag nilor pe pagina dumneavoastră Web").

Deoarece nu este nici o posibilitate de a spune unde poate apărea un document bazat pe un șablon, se folosește un URL `src`, care este relativ la rădăcina documentului paginii Web.

SUGESTIE

Atributele `height` și `width` ale imaginii sunt identice cu dimensiunile sale reale. Aceste atribute ajută la mărirea vitezei de afișare a paginii prin prevenirea browserului despre cât spațiu este necesar înainte ca aceasta să fie încărcată.

SUGESTIE

Imaginea flăcării poate să fie un GIF animat sau o animație Shockwave, după cum este descris în Capitolul 16, „Obiecte înglobate în HTML". Plasați conținutul special într-un director comun astfel încât cache-ul browserului să poată recunoaște fișierul indiferent pe ce pagină apare. Această metodă scurtează mult timpul de încărcare pentru paginile consecutive.

După ce manșeta pentru pagină este terminată, indicați unde va apărea conținutul. Pentru a face aceasta folosiți alt tabel, la care setările lățimii coloanelor să fie diferite, pentru a adăuga ehilibru machetei și a o face mai interesantă, după cum este arătat în Listingul 5.8 și Figura 5.7. În această arie de conținut, 25 la sută din lățime este alocată titlului, iar restul se aplică corpului de text.

Listingul 5.8. Adăugați aria de conținut prin folosirea altui tabel.

```
<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 4.0//EN">
<HTML>
<HEAD>
<TITLE>MRFD - a title</TITLE>
</HEAD>
<STYLE type="text/css">
<!--
BODY {background:white}
TABLE.banner {background:black}
H1.banner {color:white}
-->
</STYLE>
<BODY>
<TABLE class="banner" border="0" cellspacing="15" width="100%">
 <TR>
  <TD align="RIGHT" valign="BOTTOM" width="75%">
   <H1 class="banner">...Subject Title...</H1>
  <TD align="LEFT" valign="BOTTOM">
   <IMG src="/mrfd/images/fire2.jpg" HEIGHT="158" WIDTH="149">
</TABLE>
<TABLE BORDER="0" CELLSPACING="15" WIDTH="100%">
<TR>
 <TD WIDTH="25%" VALIGN="TOP">
  <H2>Topic Title
    <IMG SRC="/ClipArt/Earth/Landscp/Wildrnss.jpg" WIDTH="162" HEIGHT="122">
  </H2>
 <TD VALIGN="TOP">
  <P>Body text...</P>
</TABLE>
</BODY>
</HTML>
```

Cu cele două elemente principale la locul lor, șablonul paginii Web începe să capete conți (vezi Figura 5.7). Observați marcajul de rezervare care ține locul textului și care va înlocuit în fiecare pagină dezvoltată din acest șablon.

Acum etichetele de structură sunt la locul lor, pagina are o manșetă pentru titlul princip și tabelul este formatat pentru a primi conținut. Ultimul pas este să adăugăm câteva deta pentru a termina treaba, inclusiv o linie orizontală la sfârșitul paginii și o legătură înap către pagina de bază. Fișierul șablon complet, cu toate componentele la locul lor, poate văzut în Listingul 5.9.

Listingul 5.9. Fișierul `template.htm` include toate componentele pentru alte pagini a Departamentului de Pompieri.

```
<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 4.0//EN">
<HTML>
<HEAD>
<TITLE>MRFD - a title</TITLE>
</HEAD>
<STYLE type="text/css">
<!--
BODY {background: white}
TABLE.banner {background:black}
H1.banner {color:white}
.right {text-align:right}
-->
</STYLE>
<BODY>
<TABLE class="banner" border="0" cellspacing="15" width="100%">
 <TR>
  <TD align="RIGHT" valign="BOTTOM" width="75%">
   <H1 class="banner">...Subject Title...</H1>
  <TD align="LEFT" valign="BOTTOM">
   <IMG src="/mrfd/images/fire2.jpg" HEIGHT="158" WIDTH="149">
</TABLE>
<TABLE BORDER="0" CELLSPACING="15" WIDTH="100%">
<TR>
 <TD WIDTH="25%" VALIGN="TOP">
  <H2>Topic Title
    <IMG SRC="/ClipArt/Earth/Landscp/Wildrnss.jpg" WIDTH="162" HEIGHT="122">
  </H2>
 <TD VALIGN="TOP">
  <P>Body text...</P>
</TABLE>
<HR>
<P class="right">
<A HREF="/mrfd/index.htm"><STRONG>MRFD Home </STRONG>
 <IMG SRC="/mrfd/Images/cross.gif" WIDTH="133" HEIGHT="97" ALIGN="MIDDLE">
</A></P>
</BODY>
</HTML>
```

Fișierul complet `template.htm` ilustrează aspectul de bază al oricărei pagini Web create pentr Departamentul de Pompieri al Districtului Missoula. Deoarece chenarele tabelului sur invizibile, pot fi folosite tehnici de formatare avansate fără a distrage atenția utilizatorului.

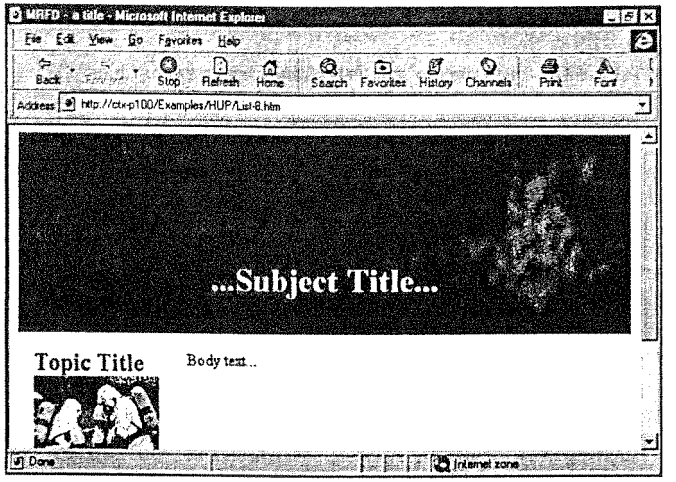


FIGURA 5.7. *Șablonul unei pagini Web după adăugarea ariei de conținut.*

OBSERVAȚIE

Se spune că o imagine valorează cât o mie de cuvinte, dar uneori vi se poate părea că durează cel puțin o mie de minute până se încarcă. Din acest motiv, majoritatea paginilor din situl Departamentului de Pompieri au numai trei poze relativ mici – una în manșetă, una sub titlul documentului și una pentru legătura către pagina de bază.

Deoarece imaginea din manșetă și cea corespunzătoare legăturii se repetă pe practic toate paginile, browserul utilizatorului ar trebui să păstreze fișierul respectiv în memoria cache pentru a mări viteza de încărcare. Singurul fișier pe care browserul are nevoie să-l încarce pentru fiecare pagină este cel de sub titlul documentului.

De ce această subliniere privind limitarea imaginii? Pentru că sunt multe ISDN-uri și modemuri de 28,8 K pe piață, fără a pomeni modemurile de 14,4 K și 9600, care sunt folosite în continuare. Această limitare este o încercare de a iuți ritmul de încărcare pentru toți utilizatorii, păstrând în continuare destule imagini pentru a face pagina interesantă.

Salvați șablonul sub numele `template.htm`. De fiecare dată când aveți nevoie de o nouă pagină, puteți pune o copie a acestui fișier în directorul respectiv și să o redenumiți. Apoi, când o deschideți pentru editare, puteți înlocui marcajul de rezervare cu conținutul dorit. Deoarece este un fișier utilitar, care nu este destinat afișării, este păstrat pe unitatea locală, nu pe situl Web.

Crearea unei pagini reale cu ajutorul șablonului

După ce este creat șablonul, faceți copii care vor fi apoi modificate pentru a crea pagini reale pentru utilizatorii dumneavoastră. Prima copie este destinată să fie una dintre paginile de siguranță ale districtului. În primul rând redenumiți-o, iar apoi adăugați conținutul. Tot ce aveți de făcut este să schimbați informația din eticheta `<TITLE>` și să adăugați conținutul corespunzător în corp. (Vezi Listingul 5.10 și Figura 5.8).

Listingul 5.10. Numai conținutul etichetei `<TITLE>` și al corpului au fost schimbate în această pagină bazată pe template.htm.

```
<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 4.0//EN">
<HTML>
```

(continuare)

Listingul 5.10. Continuare

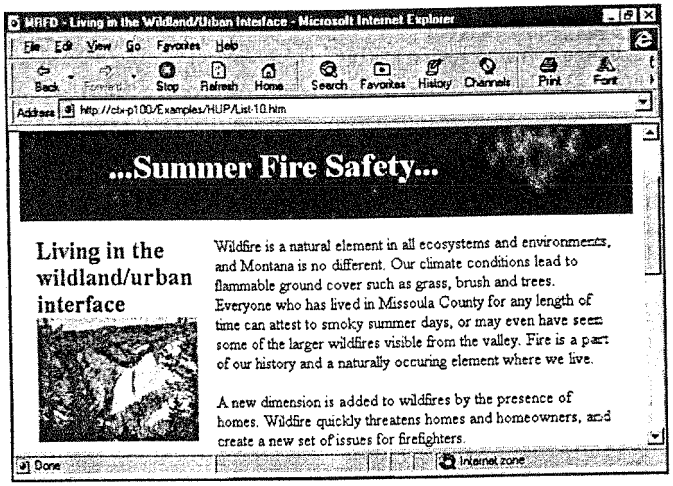
```
<HEAD>
 <TITLE>MRFD - Living in the Wildland/Urban Interface</TITLE>
</HEAD>
<STYLE type="text/css">
<!--
BODY {background:white}
TABLE.banner {background:black}
H1.banner {color:white}
.right {text-align:right}
.maroon {color:maroon}
-->
</STYLE>
<BODY>
<TABLE class="banner" BORDER="0" CELLSPACING="15" WIDTH="100%">
 <TR>
  <TD ALIGN="RIGHT" VALIGN="BOTTOM" WIDTH="75%">
   <H1 class="banner">...Summer Fire Safety...</H1>
  <TD ALIGN="LEFT" VALIGN="BOTTOM">
   <IMG SRC="/mrfd/images/fire2.jpg" HEIGHT="158" WIDTH="149">
</TABLE>
<TABLE BORDER="0" CELLSPACING="15" WIDTH="100%">
 <TR>
  <TD WIDTH="25%" VALIGN="TOP">
   <H2> Living in the wildland/urban interface
   <IMG SRC="/mrfd/images/Wildrnss.jpg" WIDTH="162" HEIGHT="122">
   </H2>
  <TD VALIGN="TOP">
  <P>Wildfire is a natural element in all ecosystems and environments, and
   Montana is no different. Our climate conditions lead to flammable ground cover
   such as grass, brush and trees. Everyone who has lived in Missoula County for
   any length of time can attest to smoky summer days, or may even have seen some
   of the larger wildfires visible from the valley. Fire is a part of our history
   and a naturally occuring element where we live.</P>
  <P>A new dimension is added to wildfires by the presence of homes. Wildfire
   quickly threatens homes and homeowners, and create a new set of issues for
   firefighters.</P>
  <P><SPAN class="maroon">Missoula Rural Fire District</SPAN> has firefighters
   trained for containing fires in the wildland/urban interface. But compared to
   a wildfire, they are very limited in manpower and equipment, and in the
   distances and terrain they protect. The first homes to burn in the recent
   Pattee Canyon fire were lost while firefighters were still enroute to the
   scene.</P>
  <P>If you're in an area which has a potential for wildfire, you can call for a
   representative from <SPAN class="maroon">Missoula Rural Fire District</SPAN>
   to come to your home and make
  <A HREF="/mrfd/Seasonal/Summer/Wildfire_Tips.html"> suggestions for creating
   defensible space</A> around your structures. <CITE>Defensible space</CITE> is
   an <DFN>area of reduced fuel</DFN> which gives your home greater odds of
   surviving a fire. This is a service provided free of charge to district
   homeowners. For more information, call 549-6172 or
    <A HREF= "mailto:mrfd@montana.com">send us a note</A>.</P>
</TABLE>
<HR>
<P class="right">
```



```
<A HREF="/mrfd/index.htm"><STRONG>MRFD Home</STRONG>
 <IMG SRC="/mrfd/Images/cross.gif" WIDTH="113" HEIGHT="97" ALIGN="MIDDLE">
</A></P>
</BODY>
</HTML>
```

Mai multe caracteristici au fost introduse în cadrul conținutului. În primul rând, orice referire la „Missoula Rural Fire District" a fost evidențiată cu maro pentru a reliefa subtil cine furnizează informația. Cu roșu sau roz s-ar evidenția mai tare, dar ar și putea distrage atenția utilizatorului care încearcă să citească. Această folosire a etichetei `<SPAN>` este acoperită în Capitolul 18.

FIGURA 5.8. *Produsul final care folosește șablonul drept punct de pornire a fost creat în doar câteva minute.*



Apoi, sunt introduse în text două hiperlegături, folosind sintaxa din Capitolul 13, „Hiperlegături și ancore". Prima este o legătură către o pagină înrudită, `/mrfd/Seasonal/Summer/Wildfire_Tips.htm`. Pentru mai multe informații despre acest subiect, utilizatorii pot executa clic pe hiperlegătură. A doua este o legătură de e-mail, `mailto:mrfd@montana.com`. Dacă utilizatorul dorește să își manifeste reacția, poate face acest lucru imediat, în loc să caute o pagină specială prin tot situl.

Altă caracteristică este folosirea etichetelor logice pentru citate și definiții. Eticheta `<CITE>` este folosită pentru a accentua termenul *defensible space*, iar eticheta `<DFN>` a fost folosită pentru a marca definiția în aceeași propoziție. În majoritatea browserelor, aceste etichete seamănă între ele – apar scrise cu caractere italice. Aceste două etichete, la fel ca alte etichete logice, indică scopul sau rolul unui text, făcând pagina mai ușor de editat ulterior.

Această pagină este simplă, dar reprezintă majoritatea conținutului de pe sit. Timpul total pentru crearea sa – inclusiv copierea și redenumirea șablonului, tastarea textului și găsirea unei imagini corespunzătoare pentru a fi folosită sub titlul documentului – nu a depășit 10 minute.

O pagină cu o listă

Următoarea pagină, creată cu `template.htm`, este un catalog al departamentelor de pompieri care fac parte din district. După ce copiați șablonul, schimbați titlul principal din manșetă și titlul paginii și adăugați lista pentru corpul de text. (Vezi Listingul 5.11 și

Figura 5.9). Adăugați și un set de liste imbricate. Mai multe detalii despre adăugarea listelor se găsesc în Capitolul 8, „Folosirea listelor pentru organizarea informațiilor".

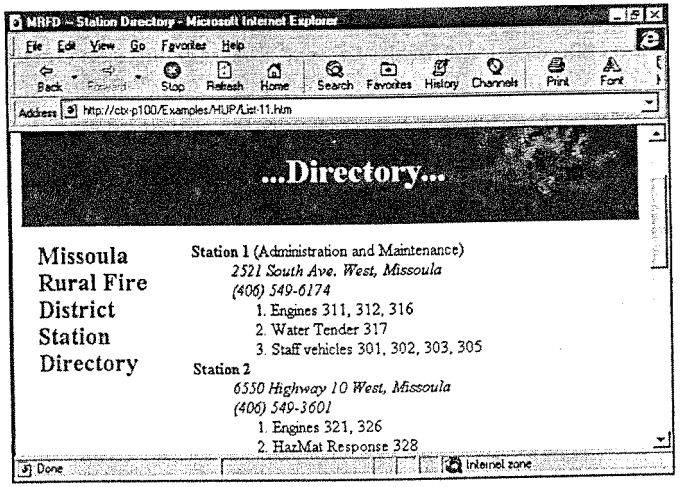
Listingul 5.11. Lista exterioară din acest set este o listă de definiții care include o listă ordonată ca parte a fiecărei definiții (<DD>).

```
<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 4.0//EN">
<HTML>
<HEAD>
 <TITLE>MRFD -- Station Directory</TITLE>
</HEAD>
<STYLE type="text/css">
<!--
BODY {background:white}
TABLE.banner {background:black}
H1.banner {color:white}
.right {text-align:right}
.maroon {color:maroon}
-->
</STYLE>
<BODY>
<TABLE CLASS="banner" BORDER="0" CELLSPACING="15" CELLPADING="1" WIDTH="100%">
 <TR>
  <TD ALIGN="RIGHT" VALIGN="BOTTOM" WIDTH="75%">
   <H1 class="banner">...Directory...</H1>
  <TD ALIGN="LEFT" VALIGN="BOTTOM">
   <IMG src="/mrfd/images/fire2.jpg" HEIGHT="158" WIDTH="149">
</TABLE>
<TABLE BORDER="0" CELLSPACING="10" CELLPADDING="5" WIDTH="100%">
 <TR>
  <TD WIDTH="25%" VALIGN="TOP">
   <H2>Missoula Rural Fire District Station Directory</H2>
  <TD WIDTH="75%" VALIGN="TOP">
   <DL>
    <DT><STRONG>Station 1</STRONG> (Administration and Maintenance)
    <DD><EM>2521 South Ave. West, Missoula<BR>
      (406) 549-6174</EM>
     <OL>
      <LI>Engines 311, 312, 316
      <LI>Water Tender 317
      <LI>Staff vehicles 301, 302, 303, 305
     </OL>

    <DT><STRONG>Station 2</STRONG>
    <DD><EM>6550 Highway 10 West, Missoula<BR>
      (406) 549-3601</EM>
     <OL>
      <LI>Engines 321, 326
      <LI>HazMat Response 328
      <LI>Staff vehicle 304
     </OL>
    </DD>

    <DT><STRONG>Station 3</STRONG>
    <DD><P><EM>Closed June 1995</EM></P>
```



```
    <DT><STRONG>Station 4</STRONG>
    <DD><EM>9480 Highway 10 East, Bonner<BR>
      (406) 258-6061</EM>
     <OL>
      <LI>Engines 341, 346
      <LI>Water Tender 347
      <LI>Truck 348
     </OL>

    <DT><STRONG>Station 5</STRONG>
    <DD><EM>12221 Highway 93 South, Lolo<BR>
      (406) 273-2551</EM>
     <OL>
      <LI>Engines 351, 356
      <LI>Water Tender 357
     </OL>
    </DD>

    <DT><STRONG>Station 6</STRONG>
    <DD><EM>8455 Mullan Road, Missoula<BR>
      (406) 542-0366</EM>
     <OL>
      <LI>Engines 361, 366
     </OL>
   </DL>
</TABLE>
<P class="right"><A HREF="/mrfd/index.htm">
<STRONG>MRFD Home </STRONG>
<IMG SRC="/mrfd/Images/cross.gif" WIDTH="133" HEIGHT="97" ALIGN="MIDDLE">
</A></P>
</BODY>
</HTML>
```

FIGURA 5.9.
În paginile acestui catalog, etichetele listelor de definiții sunt folosite când sunt necesare informații suplimentare pentru fiecare intrare din listă.



Fiecare intrare din listă conține două părți: numele și adresa departamentului și aparatele aflate în dotare. Singura listă disponibilă în HTML, care se potrivește unei relații termen-definiție, este lista de definiții. Lista aparatelor este creată prin folosirea unei liste numerotate în cadrul definiției.

Rezumat

Acest capitol v-a oferit un exemplu despre cum o organizație folosește HTML 4.0 pentru a-și construi paginile Web. Procesul a început cu experimentarea celor patru elemente de bază ale designului: titluri, text, imagini și spațiu. Aceste elemente au condus către un design simplu, care se potrivește nevoilor Departamentului de Pompieri și a devenit baza unui șablon al unei pagini Web.

Pornind de aici, au fost adăugate paginii elementele HTML – hiperlegături, mărimea și culoarea fonturilor, liste și imagini inline. Rezultatul este un aspect uniform al sitului Web care nu solicită mult timp pentru implementare. Această caracteristică este foarte folositoare deoarece situl este întreținut de oameni pentru care timpul este foarte valoros.

Este posibilă folosirea acestei combinații de grafică, tabele și cadre pentru a construi designuri de pagini mai elaborate și mai imaginative, dar scopul acestor exemple este să vedeți ce se poate face rămânând înıtre limitele stabilite de standardul elementelor HTML. Partea a IV-a, „Tehnici de prezentare", acoperă caracteristicile extinse HTML ale designului dincolo de ceea ce v-am arătat în această secțiune. Partea a VII-a se concentrează pe design, arătându-vă cum să profitați mai bine de capacitatea HTML-ului de a vă transmite mesajul către publicul nerăbdător.

PARTEA a II-a

Elemente de structură și navigație ale HTML-ului 4.0

Capitolul 6
Structurarea textului

de Rick Darnell

În acest capitol:

- **Declarația `<!DOCTYPE>` 68**
- **Fixarea limitelor cu `<HTML>` 69**
- **Elementul `HEAD` 70**
- **Elementul `BODY` 77**
- **Elementul `STYLE` 79**
- **Elementul `SCRIPT` 79**

După cum ați putut vedea în Partea I a acestei cărți, „Introducere în HTML 4.0", documentele HTML sunt create prin combinarea unor coduri de marcare numite *etichete*. Acestea definesc structura documentului și furnizează cadrul în care se află conținutul, alcătuit din text și imagini sau având o natură specială.

Elementele discutate în acest capitol se numesc elemente *structurale*, deoarece scopul lor principal este să creeze forma documentului. Aceste elemente conțin documentul însuși, secțiunile antet și corp, titlurile paginilor și alți identificatori ai documentului.

OBSERVAȚIE

De fapt, toate etichetele HTML sunt structurale prin scopul lor de a defini relațiile dintre elemente în cadrul documentului. Acest anume set se numește *structural* datorită faptului că elementele în cauză nu sunt intenționate să afecteze ceea ce se vede pe ecran. Toate celelalte etichete controlează aspectul paginii sub o formă sau alta.

Cel mai ușor mod de a ilustra scopul și forma etichetelor structurale este să începem cu o pagină, arătată în Listingul 6.1, fără nici o structură.

Listingul 6.1. Acest document HTML nu are nici o formă structurală HTML.

```
Missoula Rural Fire District Home Page
Welcome to the MRFD Home Page
```

(continuare)

Listingul 6.1. Continuare

```
Thank you for visiting our home page. From here, you can link to several
destinations, including fire safety information, district budget, minutes from
board meetings and upcoming training.
```

În cazul în care codul din Listingul 6.1 ar fi afișat pe un browser, ar arăta asemănător cu pagina din Figura 6.1. Observați lipsa totală a organizării. Aceasta ar putea genera o eroare pe un browser Web nonstandard, deoarece tipul conținutului nu este identificat.

Un design bun al unei pagini începe cu o bună organizare, aceasta fiind furnizată de etichetele structurale.

ATRIBUTE COMUNE

Majoritatea elementelor din acest capitol au anumite atribute în comun. În loc să le prezentăm pentru fiecare element în parte, le vom scrie aici și vom face referiri la ele când este cazul. Primul set de atribute se numesc atribute principale:

id: Această valoare servește ca identificator în cadrul documentului; el trebuie să fie unic, deci diferit de orice alt ID în cadrul documentului.

title: Acest atribut furnizează informații adiționale despre element și poate fi folosit sau ignorat, după cum dorește browserul.

class: Foile de stil folosesc acest atribut pentru a aplica un stil unui set de elemente asociate cu această clasă. Vezi Capitolul18, „Utilizarea foilor de stil în cascadă", pentru mai multe informații.

style: Acest atribut este folosit pentru a introduce informații inline despre stilul apariției actuale a elementului. Pentru mai multe informații vezi Capitolul 18.

Următoarele două sunt atribute internaționale (vezi Capitolul 40, „Internaționalizarea limbajului HTML"):

lang: Acest atribut specifică limbajul primar al textului din document, având valoarea prestabilită unknown (necunoscut). Pentru mai multe informații consultați „Tags for the Identification of Languages" („Etichete pentru identificarea limbajului"), disponibil la `ftp://ds.internic.net/rfc/rfc1766.txt`.

dir: Direcția unui text slab sau neutru, care poartă valoarea `LTR` (de la stânga la dreapta) sau `RTL` (de la dreapta la stânga). Browserul determină direcția prestabilită, însă pentru engleză și limbile europene putem să presupunem cu siguranță că este `LTR`.

Următoarele sunt atributele de aliniere, care determină modul în care textul este așezat în cadrul documentului. Valorile sunt `left`, `right`, `center` sau `justify`. În mod prestabilit sunt prevzute `left` pentru paragrafe `LTR` și `right` pentru paragrafe `RTL`. Multe browsere nu recunosc `justify` (atât stânga, cât și dreapta), caz în care alinierea prestabilită are valoarea dată de atributul **dir**.

Ultimele atribute sunt cele care manevrează evenimentele – `onclick`, `ondblclick`, `onmousedown`, `onmouseup`, `onmouseover`, `onmousemove`, `onmouseout`, `onkeypress`, `onkeydown` și `onkeyup` – care sunt discutate mai detaliat în Capitolul 28, „Lucrul cu utilizatorul – Evenimente".

Declarația <!DOCTYPE>

Teoretic, eticheta `<!DOCTYPE>` ar trebui să fie prima etichetă într-un document HTML. Această etichetă comunică serverului Web cu ce are de-a face când livrează documentul; tot așa serverul informează browserul la ce se poate aștepta să găsească înăuntru.

Iată cum este folosită eticheta `<!DOCTYPE>`:

```
<!DOCTYPE HTML idString>
```

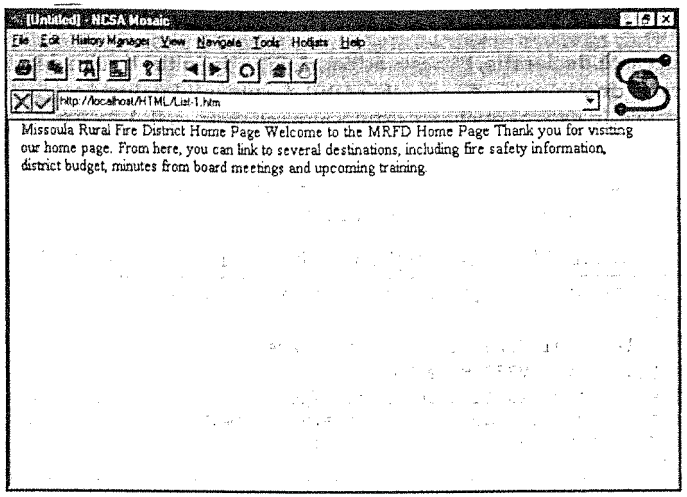



FIGURA 6.1. *Fără unele dintre elementele structurale de bază, această pagină Web nu are nici formă, nici rost.*

În această declarație, *idString* specifică versiunea de HTML folosită în document. Ca exemplu, codul următor identifică un document construit conform standardelor elaborate pentru HTML 4.0:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0//EN">
```

SUGESTIE

Puteți găsi o listă destul de completă a declarațiilor `DOCTYPE` la următoarea adresă:
`http:/ ugweb.cs.alberta.ca/~gerald/validate/lib/catalog`

În realitate, folosirea declarației `<!DOCTYPE>` este dificilă. Sunt multe versiuni de HTML în folosință, standardele se dezvoltă continuu, iar implementările browserelor continuă să schimbe regulile. Amestecul de extensii într-un document poate include etichete din HTML 2.0, 3.2 și 4.0 împreună cu extensiile personalizate de la Netscape și Microsoft. Într-o astfel de situație se întâlnesc față în față standardele universale cu regulile pieței. De fapt, majoritatea browserelor ignoră `<!DOCTYPE>`, la fel ca pe orice altă etichetă pe care nu o înțeleg, indiferent dacă face parte din standardul ales pentru document.

OBSERVAȚIE

Eticheta `<!DOCTYPE>` este de asemenea folosită sub forme particularizate de aplicații cum ar fi Microsoft FrontPage, care au nevoia de a identifica conținutul specific produsului.

După această introducere, este timpul să ne ocupăm de documentul propriu-zis.

Fixarea limitelor cu <HTML>

Prima etichetă pe care o învățăm este `<HTML>`, care este linia din document aflată imediat după `<!DOCTYPE>`. Face pereche cu `</HTML>` pentru a închide toate etichetele dintr-o pagină HTML; cele două marchează începutul și sfârșitul absolut al unui fișier.

Sintaxa etichetei `<HTML>` este:

```
<HTML [version=URL][international]>
```

```
...document and tags...
</HTML>
```

În această declarație, *URL* este adresa URL a definiției tipului de document care se aplică acestui document, iar *tags* este restul documentului dumneavoastră HTML. Elementele permise în cadrul etichetei <HTML> sunt eticheta <HEAD> urmată de un <FRAMESET> sau un <BODY>.

Folosirea etichetelor <!DOCTYPE> și <HTML> este primul pas în furnizarea unei structuri pentru primul exemplu din acest capitol; codul rezultat este prezentat în Listingul 6.2.

Listingul 6.2. Marcarea sfârșitului și începutului unui document HTML.

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0//EN">
<HTML>
Missoula Rural Fire District Home Page
Welcome to the MRFD Home Page
Thank you for visiting our home page. From here, you can link to several
destinations, including fire safety information, district budget, minutes from
board meetings and upcoming training.
</HTML>
```

Sunt câteva motive bune pentru a folosi eticheta <HTML>. În primul rând, HTML nu este singurul limbaj de marcare pe Web. Există versiuni înrudite cu HTML-ul, cum ar fi limbajul extensibil de marcare (XML), care sunt interpretate un pic diferit și câștigă din ce în ce mai mulți adepți. În al doilea rând, folosirea etichetei <HTML> este un bun stil de lucru și arată că acela care a conceput documentul avea idee despre ceea ce trebuie făcut – așa cum știți și dumneavoastră, de vreme ce citiți această carte.

OBSERVAȚIE

Etichetele <HTML> de început și de sfârșit sunt opționale și majoritatea browserelor știu ce să facă cu documentul o dată ce acesta a fost încărcat, în special dacă este conform DTD-ului și folosește elementul DOCTYPE. Totuși, unele browsere ar putea deveni confuze când li se prezintă un document fără nici o indicație clară asupra tipului său, fie DOCTYPE, fie HTML, rezultând o pagină care nu se știe cum va fi afișată sau care s-ar putea nici măcar să nu se încarce.

Aspectul Listingului 6.2 nu se va schimba, indiferent dacă se adaugă sau nu etichetele <HTML> de început și de sfârșit, dar acum limitele sale sunt stabilite, iar browserul va ști cum să interpreteze fișierul după ce acesta este încărcat.

Acum, că tipul și limitele documentului sunt marcate și identificate, este timpul să împărțim pagina HTML în două zone operaționale: antetul și corpul. Etichetele și celelalte elemente care guvernează aceste două secțiuni majore sunt explicate în cele ce urmează.

Elementul HEAD

Teoretic, fiecare document are un antet și un corp. Antetul documentului este locul unde sunt definite setările globale; este conținut între etichetele <HEAD> și </HEAD>.

Iată cum sunt folosite etichetele <HEAD>:

```
<HEAD [profile=URLlist][international]>
header content
</HEAD>
```

URLlist conține una sau mai multe adrese URL separate prin spații, care specifică fișiere cu profiluri de metadate pentru document. Profilul este discutat în detaliu în secțiunea

„Folosim <META> pentru a oferi mai multe informații", mai târziu în acest capitol. *Header content* include unul sau mai multe articole din cele șase etichete folosite exclusiv în cadrul porțiunii antet a unui document HTML. Acesta este locul preferat și pentru a introduce definițiile funcțiilor limbajului de scriptare.

Prin adăugarea acestor etichete exemplului de document HTML rezultă Listingul 6.3.

Listingul 6.3. Dezvoltăm în continuare structura cu eticheta <HEAD>.

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0//EN">
<HTML>
<HEAD>
Missoula Rural Fire District Home Page
</HEAD>
Welcome to the MRFD Home Page
Thank you for visiting our home page. From here, you can link to several
destinations, including fire safety information, district budget, minutes from
board meetings and upcoming training.
</HTML>
```

Afișat pe un browser, rezultatul s-a mai clarificat puțin, prin eliminarea primei linii. Aceasta va apărea în următoarea etichetă de antet.

Anteturile au încă o funcție importantă. Folosind prorocolul HTTP, este posibil să descărcați numai informațiile antetului. Majoritatea utilizatorilor nu cunosc această posibilitate și probabil că lor nici nu le pasă. Aceasta este folosită în primul rând de motoarele de căutare și de roboții automați, care pot să descarce un antet și să obțină unele informații de bază despre pagină – titlul, formatul fișierului, data la care a fost modificat ultima dată, cuvinte cheie – fără a pierde timpul încărcând sau vizionând întregul document.

Să atribuim paginii dumneavoastră un <TITLE>

Probabil cea mai folosită caracteristică a lui HEAD în HTML este eticheta <TITLE>. Este un element necesar în HEAD și are sintaxa:

```
<TITLE [international]>text</TITLE>
```

În această etichetă, *text* este un nume scurt, pe o singură linie, pentru documentul care este afișat în bara de titlu. Fără un titlu, majoritatea browserelor se orientează către numele de fișier HTML prestabilit. Deoarece numele fișierelor nu sunt întotdeauna deosebit de descriptive, iar câteodată sunt lungi și neatractive, este bine să furnizați un titlu pentru toate documentele HTML.

Adăugarea etichetei <TITLE> documentelor dumneavoastră HTML le întărește structura; rezultatele sunt arătate în codul din Listingul 6.4, iar pagina în Figura 6.2.

Listingul 6.4. Adăugarea titlului documentului HTML.

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0//EN">
<HTML>
<HEAD>
<TITLE>Missoula Rural Fire District Home Page</TITLE>
</HEAD>
Welcome to the MRFD Home Page
Thank you for visiting our home page. From here, you can link to several
destinations, including fire safety information, district budget, minutes from
board meetings and upcoming training.
</HTML>
```

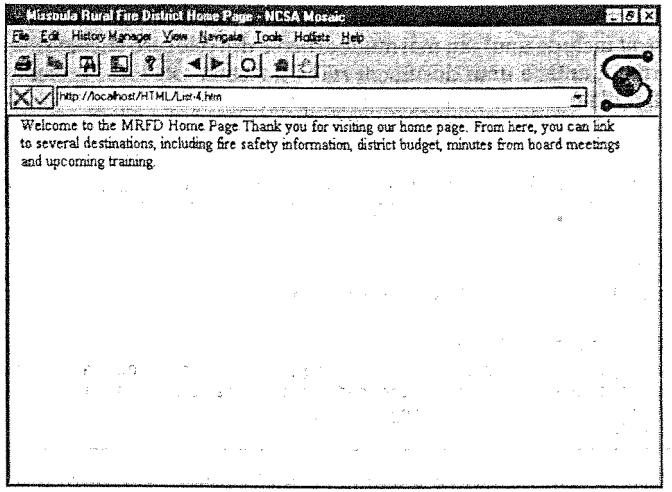


FIGURA 6.2. *Pagina Web după adăugarea unor elemente structurale și a unui titlu (arătat în bara de titlu a browserului).*

Este permis un singur titlu de pagină per document (după cum se arată în sugestia următoare), iar mărimea sa este limitată de mărimea ferestrei browserului utilizatorului. Puteți include un titlu oricât de lung, dar va fi trunchiat în bara de titlu a browserului dacă se întinde dincolo de limitele acesteia.

SUGESTIE

Pentru a face titlurile dumneavoastră mai ușor de înțeles de către utilizator, aveți grijă să fie scurte și la subiect, dar nu atât de scurte încât să fie vagi. Ca exemplu, „MRFD" nu este un titlu bun. Ce înseamnă MRFD? Unde suntem? Ce facem aici? Pe de altă parte, „Missoula Rural Fire District Home Page" este un titlu bun. Ați identificat pagina și conținutul său, exact ceea ce trebuie să știe utilizatorul.

Țineți minte că multe browsere își păstrează numele lor în bara de titlu împreună cu titlul dumneavoastră, lăsând mai puțin spațiu titlului paginii. Păstrați titlul destul de lung pentru a fi folositor, dar destul de scurt pentru a încăpea pe bară.

UN SINGUR TITLU, VĂ ROG

Odată, demult, când browserul Netscape 1.0 era cel mai tare din oraș, a fost o mică problemă cu eticheta <TITLE>. Puteați plasa o serie de etichete, după cum urmează:

```
<TITLE>M</TITLE>
<TITLE>MR</TITLE>
<TITLE>MRF</TITLE>
<TITLE>MRFD</TITLE>
<TITLE>MRFD </TITLE>
<TITLE>MRFD H</TITLE>
<TITLE>MRFD Ho</TITLE>
<TITLE>MRFD Hom</TITLE>
<TITLE>MRFD Home</TITLE>
<TITLE>MRFD Home </TITLE>
<TITLE>MRFD Home P</TITLE>
<TITLE>MRFD Home Pa</TITLE>
<TITLE>MRFD Home Pag</TITLE>
<TITLE>MRFD Home Page</TITLE>
```


Rezultatul este un titlu care se construiește treptat de-a lungul barei de titlu. Această comportare aberantă a fost rezolvată în Netscape 2.0, iar rezultatul au fost niște pagini cu un comportament ciudat. Diferite browsere interpretează titluri multiple în moduri diferite. Spre exemplu, Navigator 4.02 alege primul titlu din listă, dar Internet Explorer 4.0 folosește ultima etichetă. În cazul exemplului anterior, documentul va apărea ca având titlul „M" sau „MRFD Home Page", în funcție de browserul pe care îl preferă utilizatorul.

Morala acestei povești? Folosiți un titlu în documentul dumneavoastră pentru ca browserul să nu fie nevoit să ghicească, iar utilizatorul să nu trebuiască să se întrebe ce se întâmplă.

<BASE> și hiperlegăturile

Eticheta <BASE> controlează acțiunile hiperlegăturilor relative în corpul documentelor. În mod prestabilit, legăturile relative dintr-un document se adresează aceluiași server pe care se află pagina. Totuși, cu eticheta <BASE>, puteți specifica legăturile relativ la altă locație, care poate fi un alt director pe gazdă sau un server complet diferit. (Pentru mai multe informații despre hiperlegături și ancore, vezi Capitolul 13, „Hiperlegături și ancore").

Eticheta <BASE> necesită un singur atribut, HREF, cu sintaxa următoare:

```
<BASE href="protocol://servername/path/"
[target=targetFrame]>
```

URL-ul din atributul href specifică un URL absolut, necesar în rezolvarea hiperlegăturilor, iar *targetFrame* identifică un cadru care va primi conținutul hiperlegăturii.

OBSERVAȚIE

În eticheta <BASE>, *protocol* este un standard valid de comunicații pe Internet, la fel ca HTTP; *servername* este numele unui server sau o adresă, cum ar fi www.wossamotta.edu sau 89.123.32.21; iar *path* este orice mapare adițională de pe server. Calea (path) este o valoare opțională. Dacă este inclusă în URL, atunci se referă la serverul gazdă.

Spre exemplu, o pagină pentru o instituție educațională se află pe server la www.frostbitefalls.org. Pagina este copiată direct din serverul universității la www.wossamotta.edu și plasată la noua locație. Nici o legătură relativă nu va mai funcționa, deoarece ele fac referire la locația originală.

Când un utilizator accesează o legătură pentru a vedea programul meciurilor de fotbal, browserul ia conținutul legăturii, /football/schedule.html și îl combină cu gazda, www.frostbitefalls.org, pentru a crea http://www.frostbitefalls.org/football.schedule.html. Dar, din păcate, Frostbite Falls nu are acest director, iar utilizatorul primește înfricoșătorul mesaj de eroare 404 Not Found.

Eticheta <BASE> rezolvă problema în felul următor:

```
<BASE HREF=http://www.wosamotta.edu>
```

Acum, când un utilizator accesează legătura pentru pagina de fotbal, informația de bază este substituită cu informația gazdei, iar rezultatul va fi http://www.wossamotta.edu/football.schedule.html. Acum informația căutată este găsită ușor.

Arătarea relațiilor cu <LINK>

Eticheta <LINK> există încă din primele zile ale HTML-ului, deși nu a fost încă acceptată sau implementată pe multe browsere. În definiția sa curentă, <LINK> furnizează o metodă independentă din punct de vedere al mediului și al platformei de a defini relațiile dintre

pagina HTML actuală și alte documente și resurse. Toate browserele cu priză la public, cu excepția lui Mosaic 3.0, ignoră în continuare această etichetă.

Teoretic, <LINK> este folosită pentru a crea bare cu instrumente sau meniuri specifice documentului, în vederea controlării modul în care sunt conectate colecțiile de fișiere HTML când sunt tipărite, și pentru a conecta resursele asociate, cum ar fi foile de stil și scripturile. În plus față de atributele principale, internaționale și de evenimente, mai sunt încă cinci alte atribute folosite împreună cu LINK:

href Adresa URL, a documentului legat. Este furnizată în formatul standard URL, dar de obicei este un URL aproximativ.

rel Valoarea acestui atribut indică relația dintre documentul actual și documentul la care se face referire în href. Printre valorile folosite se numără Contents, Index, Glossary, Copyright, Next, Previous și Help. Ca exemplu, linia următoare anunță că actualul document are un glosar care poate fi găsit în fișierul intitulat glossary.html:

```
<LINK rel=Glossary href="glossary.html">
```

Acest atribut este de asemenea folosit pentru a specifica foile de stil legate, lucru descris în Capitolul 18. Altă valoare posibilă este Alternate, care indică altă versiune a documentului. De obicei, aceasta face pereche cu atributul language pentru a specifica același document în altă traducere.

rev Acest atribut este similar cu rel, cu excepția faptului că indică o relație inversă între document și URL. Ca exemplu, linia următoare anunță că documentul curent conține un glosar pentru documentul specificat în URL:

```
<LINK rev=Glossary href="chapter1.html">
```

type Acest atribut descrie controalele de introducere pentru creare. Valorile posibile sunt: text, password, checkbox, radio, submit, reset, file, hidden, image și button. Încă nu este suportat de nici un browser important.

media Acest atribut specifică mediul pe care va fi afișată informația. Poate include un singur tip de mediu, sau o listă de tipuri separate prin virgule. Valorile posibile sunt:

- **screen**: Ieșirea prestabilită, dedicată monitorului.
- **print**: Ori un format de copie hard, ori modul de previzualizare a tipăririi pe un ecran, cum ar fi un fișier Postscript.
- **projection**: Proiecția pe un ecran de format mare.
- **braille**: Ieșirea este tradusă pentru un dispozitiv Braille.
- **speech**: Ieșirea este interpretată printr-un sintetizator de vorbire.
- **all**: Ieșirea se aplică tuturor dispozitivelor.

target Numele unui cadru actual pentru a afișa conținutul legăturii.

Acum, singura întrebuințare pentru <LINK> în Internet Explorer (3.0) sau Navigator (4.0) este specificarea foilor de stil ca fiind exterioare documentului. Mosaic (3.0) suportă folosirea etichetei <LINK> pentru a crea bare de butoane personalizate imediat deasupra documentului. (Vezi Figura 6.3).

În Capitolul 19 găsiți mai multe informații despre cum puteți folosi <LINK> cu foile de stil în cascadă.

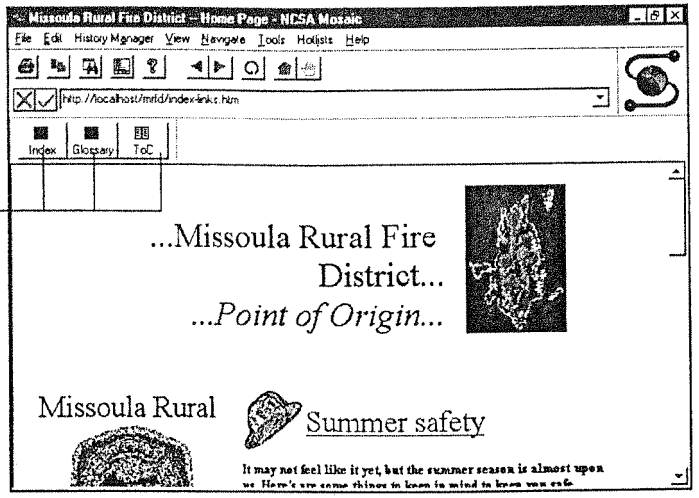


FIGURA 6.3. *NCSA Mosaic folosește etichetele <LINK> pentru a crea bare de navigare deasupra documentului.*

Butoane create cu etichete <LINK>

Folosim <META> pentru a oferi mai multe informații

Serverele Web trimit propriul lor antet împreună cu documentele HTML pentru a ajuta clienții să interpreteze documentul. Acest antet este separat și diferit de antetul la care ați lucrat în această secțiune.

La fel ca eticheta <!DOCTYPE>, eticheta <META> este folosită pentru a oferi mai multe informații despre cum trebuie manevrat documentul și deseori este folosită cu un conținut adăugat, furnizat de aplicații cum ar fi Microsoft FrontPage și Macromedia Backstage. Mai este folosită și pentru a da informații suplimentare despre document, cu ajutorul cărora motoarele de căutare pot clasifica și identifica documentul fără a-l încărca în întregime. Sintaxa este:

```
<META [http-equiv][name][content][scheme][international]>
```

Atributele http-equiv și name servesc aceluiași scop de bază, cu o singură diferență. Orice etichetă <META> care folosește http-equiv este adăugată antetului de răspuns furnizat browserului. Dacă eticheta folosește atributul name, informația este disponibilă pentru referiri în antetul documentului, dar nu este inclusă în antetul de răspuns generat de server.

Rețineți o considerație importantă pentru cazul în care folosiți atributul http-equiv. Nu trebuie niciodată să anuleze o valoare standard a antetului, cum ar fi last-modified sau contenttype. Făcând acest lucru, rezultă un conflict cu serverul, care împiedică livrarea documentului de către server sau interpretarea de către browser. Iată câteva exemple de utilizare a atributului http-equiv:

```
<META HTTP-EQUIV=refresh CONTENT=60>
<META HTTP-EQUIV=keywords CONTENT="fire department public safety">
<META HTTP-EQUIV=reply-to CONTENT= mrfd@montana.com>
```

În aceste exemple apar trei dintre cele mai folosite variabile pentru http-equiv:

- **refresh** implementează un proces numit *client-pull*, în care browserul este direcționat să reîncarce documentul după numărul de secunde specificat în content. Este folosit când documentul este actualizat periodic. Content poate lua și forma N;URL=url; N este numărul de secunde de așteptare, iar *url* este noul URL ce trebuie încărcat.

- **keywords** specifică o listă de cuvinte separate prin spații, care este folosită de motoarele de căutare pentru a clasifica documentul în vederea unei recuperări mai rapide. Multe motoare de căutare încarcă doar antetul, nu corpul, deci este importantă includerea cuvintelor cheie în eticheta <META>, pentru ca motorul să știe ce se află în documentul dumneavoastră.
- **reply-to** furnizează o adresă e-mail pentru ca utilizatorii să poată răspunde autorului sau părții responsabile pentru pagină. Afișarea sa este automat declanșată de server, care o adaugă ca pe o incluziune de partea serverului. Acest atribut nu este folosit în general.

OBSERVAȚIE

O *incluziune de partea serverului* (SSI) este o îndrumare plasată în documentul HTML care comunică serverului să insereze o bucată de text în pagină când aceasta este livrată browserului. Folosirea sa este tipică pentru lucruri ce s-ar putea schimba de la o apăsare a unei taste la alta, cum ar fi data ultimei modificări, valoarea contorului tastărilor, mărimea documentului sau adresa webmasterului. Sintaxa pentru inserarea și activarea unei SSI diferă la fiecare server Web, astfel încât este necesar să verificați documentația sau administratorul sistemului dumneavoastră pentru a primi mai multe informații.

CE ESTE CLIENT PULL?

Client pull furnizează o metodă ca paginile să se reîncarce automat după ce a trecut o anumită perioadă de timp sau ca o serie de pagini să se autoîncarce automat cu o pauză între ele. Dacă doriți ca browserul să reîncarce automat pagina curentă în patru secunde, puteți adăuga la pagina dumneavoastră HTML următoarea etichetă:

```
<META http-equiv="Refresh" content=4>
```

Dacă valoarea lui content este 0, pagina este împrospătată pe cât de repede poate să o recupereze browserul, ceea ce ar putea fi un pic cam încet dacă utilizatorul are o conectare înceată sau de calitate proastă. Cu siguranță, nu este destul de rapidă pentru nici un fel de calitate a animației.

După ce valoarea lui refresh este adăugată la http-equiv, browserul o va reâncărca la infinit. Pentru a opri acest proces, trebuie să furnizați o hiperlegătură către altă pagină fără o etichetă client pull.

Încărcarea continuă a aceleiași pagini este folositoare pentru paginile actualizate constant, cum ar fi documentele cu cotele de la bursă sau scorurile din sport. Totuși, puteți folosi client pull și pentru a încărca altă pagină. Continuarea procesului de a încărca o nouă pagină vă permite să conduceți automat utilizatorul printr-o serie de diapozitive sau de instrucțiuni. Atributul content este modificat pentru a furniza această posibilitate:

```
<META http-equiv="Refresh"
content="8;URL=http://www.mrfd.com/safety/tip2.html">
```

Adresa URL trebuie să fie completă. Nu sunt permise adrese URL relative, care consistă numai în nume de cale. Înăuntrul paginii țintă, puteți include un pointer către pagina următoare și așa mai departe. Această tehnică vă permite să încărcați orice număr de pagini consecutiv. Totuși, este o idee bună să furnizați o legătură de ieșire din procesul automat, astfel încât cititorii să nu fie obligați să privească tot spectacolul.

Atributul name este folosit pentru a furniza alte informații despre document care ar putea fi folositoare cuiva când se uită la el, dar nu trebuie neapărat să fie conținute în antet. Urmează câteva exemple:

```
<META NAME=author CONTENT="Dave Herzberg">
<META NAME=description CONTENT="Home page for MRFD">
<META NAME=copyright CONTENT="Copyright 1997, Missoula Rural Fire Dist.">
```


Atributele din aceste exemple sunt folosite mai des de editoarele HTML:

- **author** identifică persoana care a creat pagina și câteodată include numele editorului HTML.
- **description** dă o explicație pe o singură linie asupra paginii sau utilizării sale. Este folosit câteodată de motoarele de căutare pentru a furniza un sumar al paginii când afișează rezultatele căutării unui utilizator.
- **copyright** este avertismentul oficial al copyrightului pentru pagina dumneavoastră. Orice creați este subiectul protecției legilor de copyright, fie că are sau nu acest enunț. Totuși, includerea lui previne scuzele legate de ignoranță („Nu am știut") din partea eventualilor plagiatori.

Cum s-a văzut în exemplele anterioare, content este valoarea reală conținută în etichetă. Deși majoritatea browserelor și serverelor nu necesită ghilimele în jurul valorii, este bine să le folosiți. Ele înlătură ambiguitatea și posibilitatea interpretării greșite, în mod special pentru valorile care necesită mai mult de un cuvânt.

Atributul scheme este folosit pentru identificarea metodei sau a șablonului care ar trebui utilizat la evaluarea atributului content. Este inclus să ajute în încercările de a defini proprietățile bibliografiilor electronice, cum ar fi profilul Dublin Core.

Elementul BODY

La fel ca la eticheta <HEAD>, scopul principal al etichetei <BODY> este de a delimita partea principală a documentului – partea văzută de utilizator. Iată cum este folosită:

```
<BODY [bgcolor=value][onload=action][onunload=action][core][international][events]>
...document contents...
</BODY>
```

În această etichetă, bgcolor este numele culorii folosite pentru fundal în document. Culoarea poate fi de asemenea definită ca un triplet hexazecimal (vezi următoarea OBSERVAȚIE). Cuvintele document contents reprezintă un conținut HTML valid, cum ar fi text, formulare, grafică, conținut special și altele.

OBSERVAȚIE

Sunt 16 nume de culori recunoscute de HTML 4.0: black (negru), silver (argintiu), gray (gri), white (alb), maroon (castaniu), purple (purpuriu), fuchsia (roșu-purpuriu), green (verde), lime (vernil), olive (măsliniu), yellow (galben), navy (bleumarin), blue (albastru), teal (albastru-verzui) și aqua (turcoaz). Aceste nume de culori ar trebui să fie recunoscute de orice browser care acceptă culori pe platforma sa.

Puteți de asemenea specifica culoarea cu un triplet hexazecimal. Tripletul este format din trei numere hexazecimale de câte două cifre reprezentând amestecul culorilor roșu, verde și albastru. Cifrele zecimale sunt numerele 0-9, iar cifrele hexazecimale încep cu numerele 0-9 și continuă cu literele de la A la F. Un număr hexazecimal de două cifre poate avea valori de la 00 (0 în zecimale) la FF (255 în zecimale).

Astfel, culoarea roșie ar fi definită ca o saturație completă de roșu (FF sau 255 dintr-un maxim de 255) cu o absență completă a culorii verde (00 sau 0 dintr-un maxim de 255) și albastru (00, la fel ca la verde). Ca un triplet, roșu este scris FF0000. Pentru a informa HTML-ul că are de a face cu o valoare hexazecimală și nu cu un nume, este adăugat caracterul diez (#) la început, transformând valoarea completă în #FF0000. Gri va fi o combinație în proporții egale a celor trei, reprezentat ca #C0C0C0 (192 din 255 pentru fiecare culoare), iar fuchsia se obține combinând roșu pur cu albastru pur și va fi reprezentată ca #FF00FF.

În versiunea HTML anterioară, erau incluse cinci atribute care descriau valorile prestabilite pentru corpul documentului: background (imaginea de fundal), text (culoarea textului),

link (culoarea normală a legăturilor), alink (culoarea legăturii în timpul acționării acesteia) și vlink (culoarea legăturii vizitate). Aceste atribute sunt nerecomandate în HTML 4.0, datorită specificării valorilor lor cu foile de stil (vezi Capitolul 18).

SUGESTIE

Dacă este omisă, eticheta <BODY> este presupusă de browser după eticheta <HEAD>. Dacă și eticheta <HEAD> lipsește, eticheta <BODY> este presupusă imediat după eticheta <HTML> de deschidere.

Handlerele de evenimente onload și onunload sunt unice pentru eticheta <BODY>. Deși sunt discutate mai detaliat în Capitolul 28, le vom prezenta pe scurt și acum.

Evenimentul onload specifică ce script sau acțiune trebuie întreprinsă când a fost încărcat întreg conținutul documentului, dar nu a fost deocamdată redat de către browser. Acest eveniment permite autorului paginii să afișeze mesaje de avertisment sau să acționeze înainte ca utilizatorul să fi avut ocazia să interacționeze cu pagina. Rețineți că întregul conținut al documentului nu include și imaginile sau alte obiecte. Conținutul paginii pentru acest atribut este compus doar din conținutul fișierului HTML.

Evenimentul onunload este similar lui onload, cu excepția faptului că are loc în momentul în care utilizatorul a acționat în vederea părăsirii paginii curente și încărcarea alteia. Autorii paginilor pot întreprinde mai multe acțiuni ca răspuns la acest eveniment, cu o singură excepție – nu-l pot împiedica pe utilizator să plece.

Mai multe despre crearea scripturilor pentru a răspunde utilizatorului puteți găsi în Capitolul 23, „Scurtă privire asupra limbajului JavaScript" și Capitolul 24, „Folosirea limbajului VBScript".

Spargerea întregului în bucăți: <DIV> și <SPAN>

Elementul <DIV> este folosit pentru a structura un document HTML într-o serie de diviziuni. Sintaxa este următoarea:

```
<DIV [core][international][events]>Content</DIV>
```

În acest element, *Content* reprezintă una sau mai multe linii de text și alte marcaje HTML care apar în această diviziune.

Eticheta <DIV> este un element de bloc care se comportă asemănător cu eticheta <P>. Dacă eticheta <P> nu are o închidere </P>, <DIV> o închide efectiv și începe un nou paragraf. Altă caracteristică este că <DIV> nu generează pauze în paragraf înainte sau după plasarea sa.

UTILIZAREA ETICHETEI <CENTER> ÎN LOC DE <DIV align=center>

Eticheta <CENTER> este identică cu <DIV align=center>. Această suprapunere a avut loc deoarece eticheta <CENTER> a fost introdusă de Netscape cam în același timp în care elementul <DIV> era adăugat specificației HTML 3.0.

Folosirea atributului align al etichetelor <P> sau <DIV> este cel mai bun mod de a centra textul. Eticheta <CENTER> se află încă în specificația HTML 4.0 deoarece este larg folosită, deși a fost nerecomandată și însemnată pentru a fi scoasă.

Eticheta <SPAN> este asemănătoare cu eticheta <DIV>, cu excepția faptului că este mai mult folosită pentru a marca conținutul inline decât pentru a crea un bloc nou. Deoarece este un element inline, nu folosește atributul de aliniere:

```
<SPAN [core][international][events]></SPAN>
```


Etichetele <DIV> și <SPAN> sunt folosite în special la marcarea secțiunilor de text pentru formatare sau poziționare cu ajutorul foilor de stil. După cum veți descoperi în Partea a VI-a, „HTML dinamic", aceste două elemente structurale sunt importante pentru unele dintre stilurile interactive și abilitățile de poziționare din documentele dinamice.

Elementul STYLE

Elementul STYLE este introdus în document pentru setarea tuturor proprietăților care controlează înfățișarea sa, inclusiv selectarea fontului, culoarea, alinierea și chenarele. Eticheta <STYLE> este acceptată de o recomandare W3C separată despre foile de stil în cascadă.

Puteți găsi mai multe informații despre foi de stil și folosirea etichetei <STYLE> în Capitolele 17, 18, 19 și 21, care acoperă foile de stil în cascadă (CSS) și folosirea lor, foile de stil bazate pe JavaScript și alte soluții alternative la CSS.

Elementul SCRIPT

SCRIPT, un element mai puțin convențional, este folosit în două moduri. Indiferent de aplicație, indică limbajul specific de marcare folosit în document. În prezent, cele mai utilizate limbaje de scriptare pe World Wide Web sunt: JavaScript și Visual Basic Script (VBScript). Sintaxa elementului este următoarea:

```
<SCRIPT [type=MIMEtype][language=language][src=URL>
...script content...
</SCRIPT>
```

În acest element, *MIMEtype* este tipul Internet MIME al limbajului, cum ar fi text/vbscript; *language* este de obicei ori JavaScript, ori VBScript, iar *URL* identifică o sursă exterioară pentru script. Mai multe informații despre specificarea și crearea scripturilor găsiți în Capitolele 23 și 24.

De obicei, <SCRIPT> conține o secțiune de cod de programare. Dacă acest cod este plasat în antet, este interpretat înainte ca restul documentului să fie încărcat. Această metodă este des folosită pentru segmente de cod care servesc pe post de funcții, când este obligatoriu ca funcția să fie încărcată și disponibilă înainte ca vreo parte din document să aibă ocazia să o invoce.

Browserele care nu au suport pentru limbajele de scriptare probabil că nu vor recunoaște eticheta <SCRIPT>. Dacă se întâmplă acest lucru, conținutul etichetei va fi afișat ca text normal, fapt care conduce la o lectură foarte ciudată pentru utilizator. Pentru a evita această problemă, iată o soluție simplă:

```
<SCRIPT>
<!--
script stuff
-->
</SCRIPT>
```

Ca urmare a încadrării scriptului cu etichetele HTML pentru comentariu în interiorul etichetelor <SCRIPT>, un browser incompatibil va ignora conținutul scriptului. Scriptul va ignora etichetele de comentariu și va continua procesarea normală.

SUGESTIE

Iată o altă metodă de a defini un script care este acceptat de Microsoft și Netscape, dar nu este încă sancționat de standardul HTML:

```
<SCRIPT language=language for=id event=eventHandler>
...script commands for this function...
</SCRIPT>
```

(continuare)

(continuare)

În această metodă, *id* identifică atributul `id` al elementului pe pagina pe care este creată funcția și *eventHandler* este acțiunea ce activează funcția.

Ca exemplu, următoarea parte de cod creează un script care să reacționeze la un clic pe butonul `nextPage`:

```
<SCRIPT language=javascript for=nextPage event=onClick>
...script commands for this function...
</SCRIPT>
<BODY>
...HTML stuff...
<INPUT type=button value="Move on" id=nextPage>
...HTML stuff...
</BODY>
```

Rezumat

Etichetele de structură pregătesc terenul pentru restul HTML-ului și furnizează cadrul de lucru pentru acesta. Ele machează începutul și sfârșitul unui fișier, antet sau ale unor secțiuni de corp. Etichetele și atributele adiționale din antet pregătesc comportamentele de bază, definesc un titlu și furnizează informații browserului. Prin folosirea etichetei `<BODY>`, puteți seta imaginile și culorile de fundal, precum și culorile prestabilite pentru obiectele text din pagină.

Deși aceste etichete au un rol important într-un document HTML, ele nu reprezintă partea vizibilă, cu care este distractiv să te joci. Discuția despre elementele care dau scop textului dumneavoastră urmează în Capitolul 7, „Alinierea și formatarea textului".

Capitolul 7
Alinierea și formatarea textului

de Rick Darnell

În acest capitol:

- **Titlurile, de la `<H1>` până la `<H6>`** **82**
- **Bazele formatării textului** **83**
- **Formatarea textului după întrebuințarea sa** **89**

În Capitolul 6, „Structurarea textului", ați văzut cum se construiește un cadru pentru un document HTML. Structura de bază include granițele paginii HTML, un antet, un corp și multe alte lucruri pentru a furniza cadrul de lucru pentru conținutul paginii. Cu aceste câteva cunoștințe despre HTML, aveți mai mult decât este necesar pentru construirea unei pagini Web.

Puteți plasa tot textul dumneavoastră între două etichete de corp, iar utilizatorul ar putea să-l citească. Există totuși o problemă. Ar apărea ca un paragraf lung, neîntrerupt, indiferent de cum l-ați introdus sau câte rânduri ați lăsat între paragrafe. Browserele Web nu percep textul cum îl percepe o persoană. Când browserul vede sfârșitul rândului în documentul dumneavoastră, adaugă un spațiu și continuă citirea. Rezultatul este un lung șir de cuvinte pe pagină care nu este nici atractiv, nici ușor de citit.

Elementele HTML discutate în acest capitol sunt create să adauge o aliniere și o formatare de bază paginii dumneavoastră. Sunt etichete pentru titluri, adrese și citate; sunt etichete pentru începerea unui paragraf nou, pentru a începe rânduri noi și mai sunt etichete pentru adăugarea liniilor orizontale care ajută la împărțirea paginii.

Acest capitol explică elementele folosite la organizarea și formatarea documentului dumneavoastră. Veți observa că practic toate paginile de pe Web folosesc cel puțin una sau două dintre aceste etichete, iar unele dintre ele folosesc mai multe. Citiți în continuare pentru a afla cum să folosiți elementele de formatare a textului în vederea furnizării structurilor vizuale și logice paginilor dumneavoastră.

ATRIBUTE COMUNE

Majoritatea elementelor din acest capitol folosesc aceleași atribute HTML, cum ar fi cele care controlează identificarea, stilul, limbajul și tratarea evenimentelor. Deoarece atributele comune se întâlnesc peste tot în HTML 4.0 (de aceea se și numesc „comune"), veți găsi o trecere în revistă a acestora în Introducere, la începutul cărții. Puteți, de asemenea, afla mai multe informații despre aceste des folosite caracteristici și elementele asociate lor în Anexa A, „HTML 4.0 – referințe".

Între timp, în acest capitol, mă voi referi la aceste atribute comune folosind următoarele cuvinte cheie: *principale*, *internaționale* și *evenimente*.

În plus, multe elemente pot accepta alte elemente în cadrul granițelor lor. Acestora li se va spune *elemente inline* și includ următoarele opțiuni:

Elemente `font`: Printre aceste elemente se numără etichetele pentru bold (`<B>`) și italic (`<I>`) și sunt discutate în Capitolul 17, „Stil simplu cu text".

Elemente `phrase`: Printre aceste elemente se numără etichete cum ar fi evidențiere (`<EM>`), cod (`<CODE>`) și adresă (`<ADDRESS>`) și sunt discutate în acest capitol.

(continuare)

(continuare)

Elemente `special`: Această colecție este o amestecătură de elemente care include următoarele: miniaplicații și obiecte (Capitolul 16, „Obiecte înglobate în HTML"), imagini (Capitolul 11, „Adăugarea imaginilor pe pagina dumneavoastră Web"), ancore (Capitolul 13, „Hiperlegături și ancore"), hărți imagine (Capitolul 14, „Crearea meniurilor cu hărți imagine"), scripturi (Capitolul 23, „Scurtă privire asupra limbajului JavaScript" și 24,"Folosirea limbajului VBScript"), fonturi, superscripturi și subscripturi (Capitolul 17), salturi la rând nou și citate scurte (în acest capitol) și cadre inline (Capitolul 20, „Crearea cadrelor").

Elementele `form`: Acest set de elemente este folosit pentru a adăuga elemente unui formular, inclusiv butoane, câmpuri de text și casete de validare.

Titlurile, de la <H1> până la <H6>

Un articol dintr-un ziar nu începe cu paragraful principal; începe cu un titlu. Cu aceasta începe prezentul capitol – cu etichetele de formatare a textului. Cele șase stiluri HTML de titlu sunt o metodă de a arăta gradul de importanță între diferite părți ale paginii dumneavoastră, precum această carte folosește niveluri diferite de titlu pentru a organiza vizual informațiile.

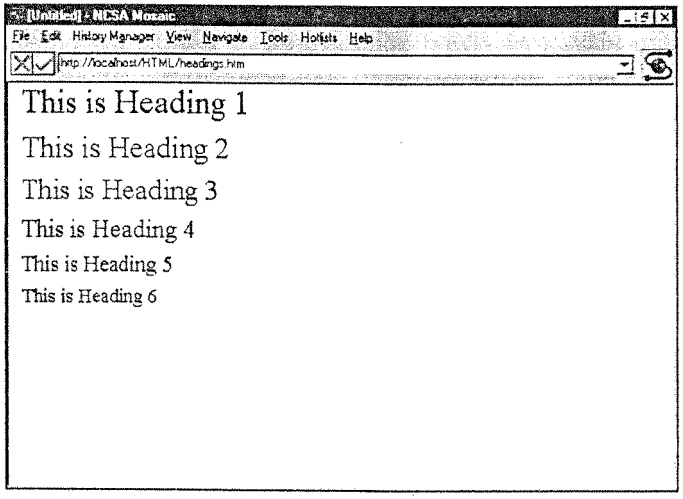
Sintaxa este următoarea:

```
<Hn [core][international][events]>Heading Text</Hn>
```

Aici, *n* este un număr întreg de la 1 la 6 care indică nivelul de titlu folosit, unde 1 este primul ca importanță, iar 6 este ultimul. Toate atributele sunt opționale.

Deși browserele folosesc mărimi și stiluri de caractere diferite pentru cele șase niveluri de titlu, fiecare browser urmează regula de bază de a atribui stilul cu cele mai mari și mai îngroșate caractere lui <H1> și caracterele cele mai mici și mai subțiri lui <H6>. (Vezi Figura 7.1).

FIGURA 7.1. *Stilurile titlurilor din HTML arată un nivel descendent al evidențierii și al importanței.*



SUGESTIE

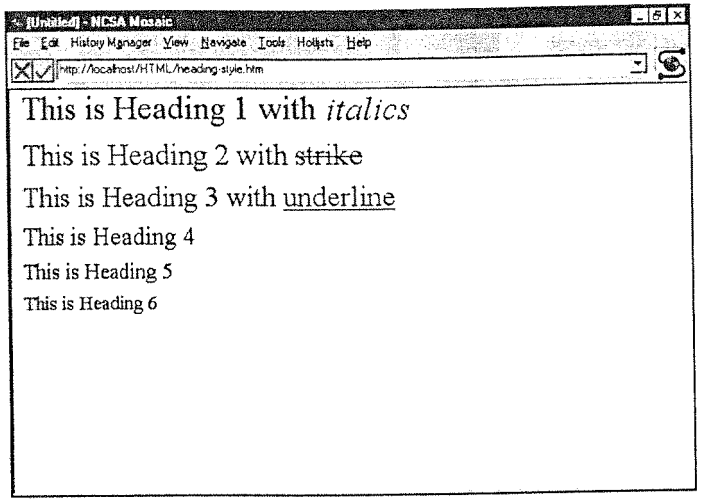
În cazul unor browsere, titlurile de pe ultimul nivel sunt chiar mai mici decât textul corpului. Rezultatul arată un pic ciudat.

Titlurile pot conține orice element inline între granițele lor. Pentru a scoate un titlu în evidență îi puteți seta stilul (bold, italic sau subliniat), tipul de caractere și mărimea, după cum se arată în acest exemplu:

```
Next on our site is <H1>Reflections on <I>The Devils
of Loudun</I></H1> followed by pictures of my cat.
```

Este posibilă schimbarea stilului oricărui titlu, indiferent de nivel, pentru a controla mai bine aspectul acestuia. (Vezi Figura 7.2).

FIGURA 7.2. *Puteți folosi formatarea fizică pentru a controla tot sau doar o parte din textul unui titlu.*



În Figura 7.2, observați că browserul manevrează textul înainte și după etichetele titlului. Un titlu este întotdeauna plasat pe propriul său rând, chiar dacă este plasat inline în alt material. Pentru titlu este început un nou paragraf, iar materialul care îl urmează este plasat pe un rând nou.

Bazele formatării textului

Acum, că am potrivit titlurile, este timpul să vă îndreptați atenția către restul documentului. Următorul pas este să începem să împărțim textul în paragrafe. Listingul 7.1 pare a fi o pagină Web spartă în paragrafe logice, dar priviți Figura 7.3 pentru a vedea cum îl interpretează un browser.

Listingul 7.1. Se încearcă folosirea unor retururi hard pentru a forța fiecare instrucțiune pe propriul său rând.

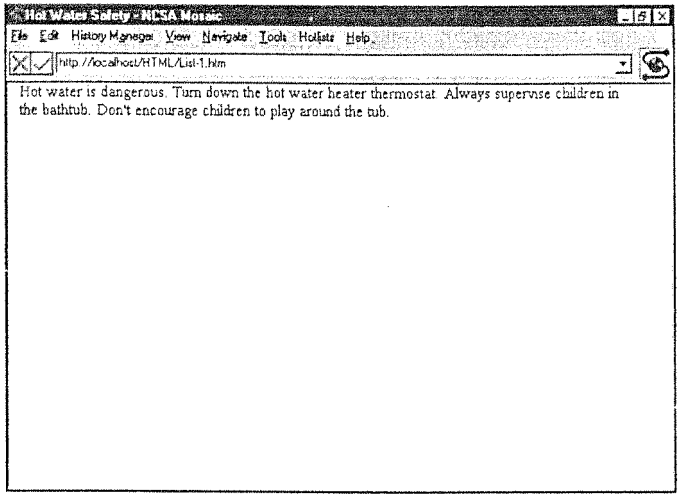
```
<HTML>
<HEAD>
<TITLE>Hot Water Safety</TITLE>
</HEAD>
<BODY>
Hot water is dangerous.

Turn down the hot water heater thermostat.

Always supervise children in the bathtub.

Don't encourage children to play around the tub.
```

(continuare)

Listingul 7.1. Continuare

```
</BODY>
</HTML>
```

FIGURA 7.3. *Browserul a ignorat rândurile de pauză în plus și a prelucrat rândurile la un loc, într-un paragraf lung.*



Evident că nu acesta este rezultatul pe care îl dorise designerul. Următoarele trei elemente oferă însă metode pentru a face pagina să se conformeze modului în care ar trebui să arate.

Un nou paragraf: <P>

Prima metodă de a împărți textul în paragrafe este folosirea etichetei de paragraf: `<P>`. Sintaxa arată așa:

```
<P [core][international][events]>...Text...</P>
```

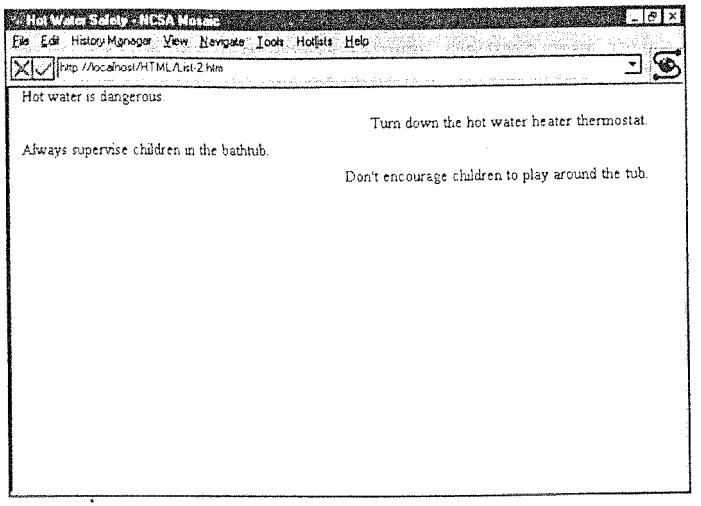
Text este un rând sau un paragraf de text, care ar trebui să înceapă pe un rând nou și să formeze un tot. Eticheta de închidere este opțională (poate fi omisă). Dacă este folosită cu o foaie de stil, este necesară eticheta de închidere pentru a aplica proprietățile foii de stil.

Folosirea acestei etichete în listingul precedent rezultă în codul arătat în Listingul 7.2 și în pagina din Figura 7.4.

Listingul 7.2. Definirea fiecărui rând ca fiind propriul său paragraf.

```
<HTML>
<HEAD>
<TITLE>Hot Water Safety</TITLE>
</HEAD>
<BODY>
<P>Hot water is dangerous.</P>

<P align=right>Turn down the hot water heater thermostat.</P>

<P>Always supervise children in the bathtub.</P>

<P align=right>Don't encourage children to play around the tub.</P>
</BODY>
</HTML>
```


FIGURA 7.4. *Eticheta <P> marchează fiecare propoziție ca fiind propriul său paragraf, deși rândul suplimentar de pauză este în continuare ignorat.*



Observați comportamentul alinierii. Un paragraf nou încalcă orice setare a alinierii din paragraful anterior. De asemenea, dacă alinierea nu este specificată, se folosește setarea prestabilită (la stânga).

Elementele inline pot fi folosite în etichetele `<P>` fără a cauza începerea unui rând nou. Elementele la nivelul blocului, cum ar fi formularele și titlurile, forțează sfârșitul paragrafului și începerea unui rând nou.

Un nou rând:

O etichetă de întrerupere de rând, `<BR>`, este asemănătoare cu o etichetă de paragraf, dar se comportă un pic diferit. Începe un rând nou în cadrul paragrafului curent, dar nu începe un paragraf nou. Sintaxa este:

```
<BR [clear=left|all|right|none][core]>...Text...
```

Text este materialul care ar trebui să apară pe rândul următor, iar atributul opțional *clear* definește cum ar trebui așezat materialul în jurul imaginilor din text. Nu este folosită o etichetă de închidere.

Folosită în exemplul anterior, această etichetă creează același efect de bază precum folosirea unei etichete de paragraf și păstrează tot textul în cadrul aceluiași paragraf. (Vezi Listingul 7.3 și Figura 7.5).

Listingul 7.3. Fiecare linie este începută cu o etichetă de întrerupere de rând în locul celei de paragraf nou.

```
<HTML>
<HEAD>
<TITLE>Hot Water Safety</TITLE>
</HEAD>
<BODY>
<P>Hot water is dangerous.

<BR>Turn down the hot water heater thermostat.

<BR>Always supervise children in the bathtub.
```

(continuare)

Listingul 7.3. Continuare

```
<BR>Don't encourage children to play around the tub.</P>
</BODY>
</HTML>
```

În Figura 7.5, observați că rândurile sunt un pic mai apropiate, deoarece ele fac parte din același paragraf.

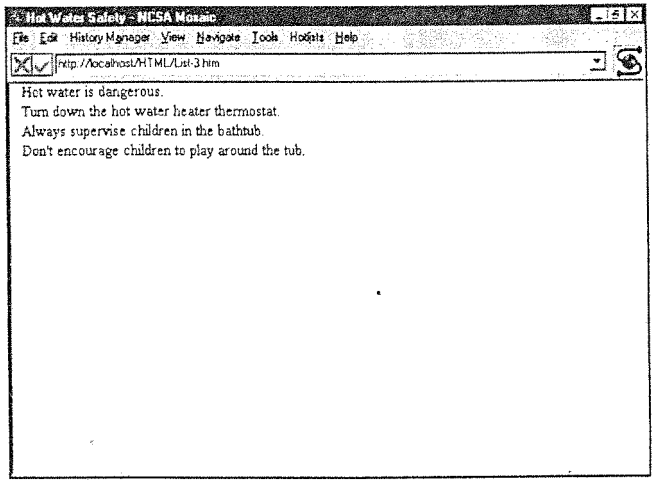


FIGURA 7.5. *Crearea unei pagini cu fiecare propoziție pe propriul său rând.*

Marea diferență pe care o veți observa între întreruperile de rând și paragrafele noi este spațierea rândurilor. (Vezi Figura 7.6). O întrerupere de rând folosește aceeași spațiere ca în cazul în care rândul tocmai a fost desfășurat din cel anterior. Un paragraf nou de obicei folosește un plus de jumătate de rând de spațiu, deoarece primul rând dintr-un paragraf nu este indentat.

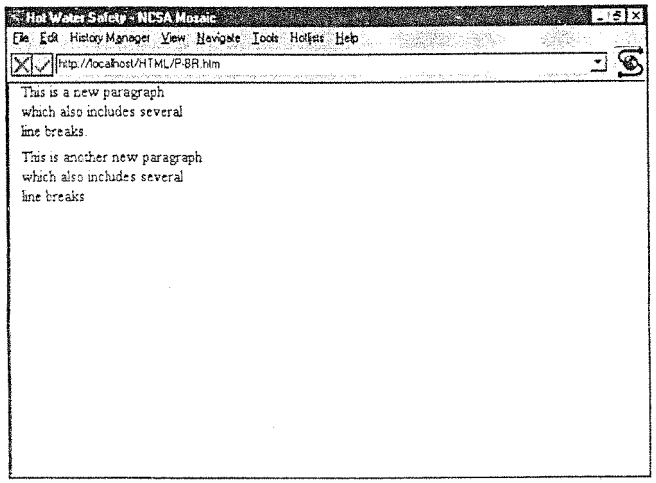


FIGURA 7.6. *Diferența principală dintre o întrerupere de rând și un paragraf nou este spațiul lăsat între rânduri.*

Orice aliniere sau altă formatare a textului care este stabilită înainte de întreruperea rândului în cadrul aceluiași paragraf este transferată noului rând.

Atributul `clear` este folosit pentru a trece dincolo de imaginile flotante, spre oricare din margini. Valoarea `left` mută textul de după întreruperea de rând dincolo de imaginea flotantă către marginea stângă (după cum se vede în Figura 7.7), iar o valoare `right` acționează la fel pentru imaginile flotante de pe partea dreaptă. După cum probabil vă așteptați, `all` face același lucru pentru imaginile flotante indiferent de margini. Alte opțiuni pentru ca textul să curgă în jurul imaginilor sunt discutate în Capitolul 11.

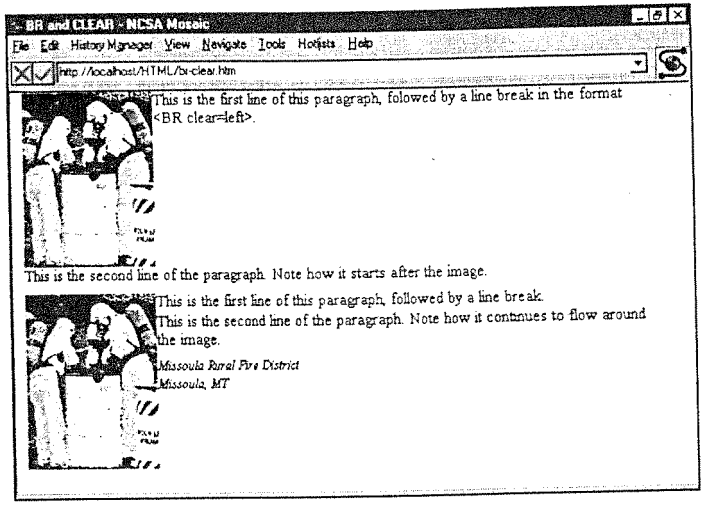


FIGURA 7.7. *Folosirea atributului `clear` pentru a permite textului să se mute dincolo de o imagine flotantă.*

Imaginile standard inline nu sunt flotante, așa că nu sunt afectate de atributul `clear`. O imagine inline, prin definiție, apare doar în rândul în care este introdusă.

Text preformatat, cu spații și cu de toate: <PRE>

Ultima metodă de a face ca cele patru rânduri ale textului să apară cum vreți dumneavoastră este să marcați toată secțiunea ca fiind text preformatat. Sintaxa arată în felul următor:

```
<PRE [width=number][core][international][events]> ...Text...</PRE>
```

`Text` este orice text, inclusiv tasta ENTER, spații și alte formatări hard. Atributele `width` comunică browserului câte caractere pe lățime să permită conținutului și îl lasă pe acesta să aleagă o mărime potrivită a fonturilor sau a indentării, astfel încât textul să fie afișat corespunzător. Atributul `width` nu este acceptat de multe browsere.

Există mai multe diferențe esențiale între această metodă și cele două etichete anterioare. Vezi Listingul 7.4 și Figura 7.8. Tot textul este încadrat de etichetele `<PRE>`, ceea ce determină browserul să afișeze tot ce este între acestea *exact așa cum se găsește*. Folosește, de asemenea un font monospațiat pentru afișare, ceea ce ajută la formatarea textului pe rânduri și coloane pentru prezentarea datelor.

Listingul 7.4. Eticheta <PRE> comunică browserului să prezinte textul exact cum a fost introdus.

```
<HTML>
<HEAD>
```

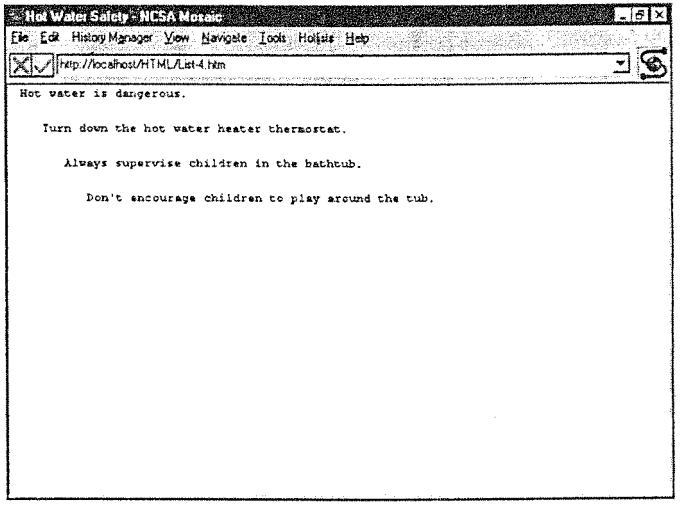
(continuare)

Listingul 7.4. Continuare

```
<TITLE>Hot Water Safety</TITLE>
</HEAD>
<BODY>
<PRE>Hot water is dangerous.

    Turn down the hot water heater thermostat.

      Always supervise children in the bathtub.

         Don't encourage children to play around the tub.</PRE>
</BODY>
</HTML>
```

Observați în Figura 7.8 că fontul monospațiat păstrează spațierea caracterelor pentru o mai precisă afișare a indentărilor și a altor formatări.

FIGURA 7.8. *Preformatarea afișează rândurile exact cum au fost tastate în fișier.*



SUGESTIE

Pentru browserele care nu sunt compatibile cu tabelele, textul preformatat este un mod eficient de a afișa un tabel compus din elemente textuale reprezentând liniile de chenar.

Fontul monospațiat ajută la păstrarea întreruperilor de rând și a spațiilor introduse de utilizator. Dacă începeți un rând nou, introduceți cinci spații și apoi începeți să tastați, exact așa va apărea textul dumneavoastră prin folosirea etichetei <PRE>. Browserele dezactivează aranjarea automată a cuvintelor pentru a se asigura că textul dumneavoastră va apărea axact așa cum l-ați tastat.

ATENȚIE!

Înainte de HTML 4.0, erau folosite alte trei etichete pentru preformatare: <XMP>, <LISTING> și <PLAINTEXT>. După lansarea HTML-ului 4.0, aceste etichete au devenit depășite. Browserele le vor mai accepta probabil o vreme, dar dacă vreți să lucrați conform standardelor HTML, este mai bine să treceți acum la eticheta <PRE>.

Formatarea textului după întrebuințarea sa

După cum am mai spus în Partea I, „Introducere în HTML 4.0", una dintre întrebuințările de bază a HTML-ului este marcarea textului astfel încât scopul și structura sa, și nu numai înfățișarea, să fie evidente. Aduceți-vă aminte că în procesarea și publicarea tradițională a textului oamenii aveau tendința să gândească în stiluri fizice – font, mărime, bold, italic și altele.

Însă HTML este o metodă de a defini structura în aceeași măsură ca înfățișarea. HTML este un limbaj de marcare simplu, folosit pentru a crea documente hipertext care să fie portabile de pe o platformă pe alta (vezi Capitolul 2,"Ce este limbajul de marcare a hipertextului?"). În cadrul acestor limitări, fiecare browser are un rol important în privința modului în care interpretează etichetele. Specificația HTML 4.0 include recomandări despre înfățișări preferate, dar browserele se pot abate de la ele, în funcție de nevoile și limitările platformelor lor.

După această scurtă recapitulare, este timpul să introducem etichetele pentru stilurile logice. Aceste etichete descriu întrebuințarea unei părți din text, în locul atributelor fizice pe care ar trebui să le aibă textul. Este cam același lucru ca definirea unui stil pentru adrese în procesorul dumneavoastră de text. De fiecare dată când aveți o adresă, aplicați stilul „adresă". Veți înceta să gândiți textul în formatare fizică o dată ce ați introdus noul stil.

Marea diferență între procesorul dumneavoastră de text și browser este că perechea browser-platformă decide cum ar trebui să vedeți textul, în loc să lase acest lucru numai la nivelul toanelor autorului.

Elementele phrase

Primul set de elemente pe care le vom discuta sunt elementele HTML phrase, folosite pentru a adăuga informații despre structură fragmentelor de text, și nu întregului paragraf sau rând.

Evidențierile simple și întărite

Câteodată trebuie să vă asigurați că utilizatorul nu scapă din vedere scopul mesajului dumneavoastră. Pentru a face aceasta, aveți nevoie să evidențiați un cuvânt sau mai multe dintr-o propoziție. HTML furnizează eticheta de evidențiere (<EM>) pentru a satisface această necesitate. Sintaxa este următoarea:

```
<EM [core][international][events]>...Text...</EM>
```

Text este cuvântul sau cuvintele pe care doriți să le evidențiați. Se recomandă utilizarea caracterelor italice pentru a reda textul evidențiat.

O variantă a etichetei de evidențiere este eticheta de evidențiere întărită (<STRONG>), care va sugera cititorului un nivel mai mare al importanței. Sintaxa arată așa:

```
<STRONG [core][international][events]>...Text...</STRONG>
```

Text este cuvântul sau cuvintele pe care doriți să le evidențiați mai puternic. Diferența dintre aceste două etichete este prezentată în Figura 7.9.

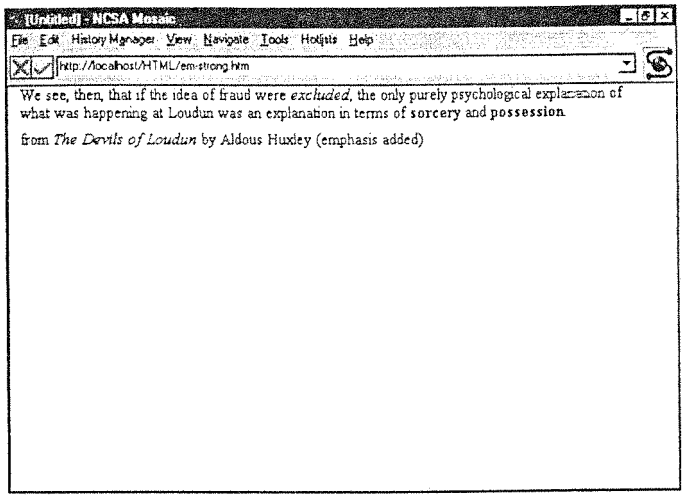
ALTE OPȚIUNI PENTRU EVIDENȚIERE

În Capitolul 17, veți vedea cum sunt folosite etichetele italic (<I>) și bold (<B>) pentru a genera același efect ca evidențierea simplă și cea întărită. Atunci de ce ar folosi cineva stilurile logice? Sunt mai lungi și nu sunt atât de clare asupra aspectului lor final.

(continuare)

(continuare)
Aduceți-vă aminte că parte din scopul HTML-ului este prezentarea structurii și scopului în cadrul documentului, iar acesta este rolul celor două etichete de evidențiere. Etichetele fizice bold și italic ar produce același efect vizual, dar folosirea etichetelor logice arată în plus intenția și este stabilă pe toate platformele – nu doar pe cele care acceptă o varietate de stiluri de caractere.

FIGURA 7.9.
Textul evidențiat este de obicei afișat cu caractere italice, iar textul întărit folosește tipul bold.



Notarea unei surse: <CITE>

Câteodată, când construiți pagini, veți dori să atribuiți faptele sursei lor sau să atrageți atenția asupra unui acronim. Încă o dată, HTML furnizează etichete care identifică cum este folosit textul și lasă detaliile aspectului său fizic browserului.

Eticheta de citat, <CITE>, este folosită pentru a identifica sursele informațiilor aflate în afara documentului curent. Este folosită cel mai des în cercetare și în paginile profesionale, dar îi puteți descoperi întrebuințări și în alte domenii, cum ar fi cronicile literare și Frequently Asked Questions (întrebări puse des). Sintaxa este următoarea:

```
<CITE [core][international][events]>Source</CITE>
```

Source este numele citatului. Redarea oficială a lui <CITE> se face în caractere italice.

Definirea termenilor: <DFN> și <ACRONYM>

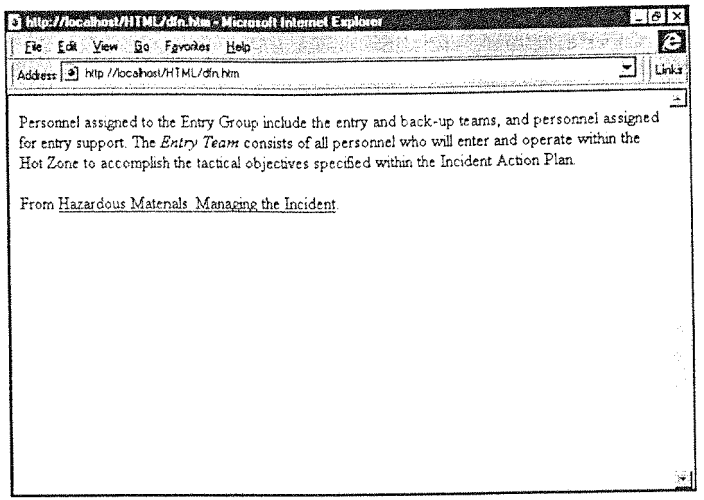
Când vă gândiți la termeni și definiții, un dicționar sau un glosar reprezintă primul lucru care vă vine în minte. HTML oferă o metodă de a prezenta acest tip de informație prin diferite tipuri de liste, care sunt discutate pe larg în Capitolul 8, "Folosirea listelor pentru organizarea informațiilor". Deși listele HTML sunt metoda obișnuită pentru prezentarea termenilor și definițiilor, mai există o variantă.

Cealaltă posibilitate de a prezenta termenul unei definiții este folosirea etichetei <DFN>. Sintaxa sa este următoarea:

```
<DFN [core][international][events]>Term</DFN>
```

Term este cuvântul definit. Este folosită la identificarea cuvintelor în cadrul unui corp de text care include definiția în aceeași propoziție. (Vezi Figura 7.10). Redarea obișnuită pentru un termen definit se face cu italice.

FIGURA 7.10.
Cuvintele „Entry Team", scrise cu caractere italice, reprezintă termenul definiției.



Eticheta <ACRONYM> este folosită pentru a evidenția amestecurile de litere și numere care par a fi atât de populare printre pasionații de calculatoare și oficialii de la guvern, cum ar fi WWW, W3C, ASCII, OSHA, DoD, IRS și altele. Se utilizează astfel:

```
<ACRONYM [core][international][events]>allTheLetters </ACRONYM>
```

Aici, *allTheLetters* este acronimul.

OBSERVAȚIE

<ACRONYM> este o etichetă nouă care nu este acceptată deocamdată pe nici un browser, deși este normal să o regăsim în următoarea versiune a Navigator-ului sau a Internet Explorer-ului. Până atunci, nu există o reprezentare sugerată sau preferată, dar putem presupune că aceasta se va face fie în italice, fie în bold.

Indicarea codului de program: <CODE> și <VAR>

Ați văzut exemple de sintaxă și folosire a codului HTML de-a lungul acestui capitol și le veți vedea în continuare pe parcursul întregii cărți. A fost folosit un stil de caractere special, monospațiat pentru a prezenta aceste exemple, astfel încât să iasă în evidență față de restul textului.

HTML are o caracteristică asemănătoare prin eticheta <CODE>, care este folosită la formatarea programului sau a altor rânduri de cod similare. Sintaxa este următoarea:

```
<CODE [core][international][events]>...CodeLine</CODE>
```

CodeLine este rândul sau rândurile de eșantioane de cod. Dacă sunt incluse rânduri multiple de cod, este de asemenea necesară eticheta de întrerupere de rând, <BR>.

Este important să observați că deși afișajul obișnuit al acestui element folosește text monospațiat ca în cazul elementului PRE, <CODE> necesită etichetele <BR> sau <P> pentru a forța începerea noilor rânduri. În afară de aspectul textului, orice este încadrat între etichetele <CODE> se comportă la fel ca orice alt text HTML.

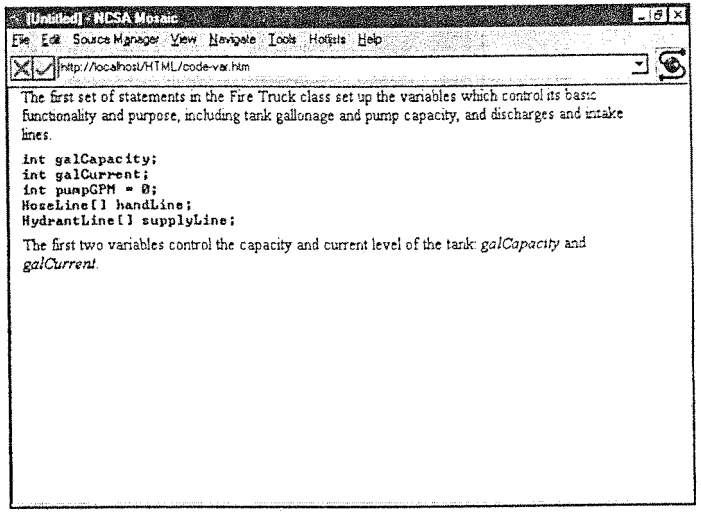
Eticheta <VAR> este des folosită împreună cu eticheta <CODE> pentru a ajuta explicarea codului. Sintaxa sa este următoarea:

```
<VAR [core][international][events]>Variable</VAR>
```

Variable este numele variabilei descrise, similar cu modul în care <DFN> lucrează cu termenii definițiilor. Reprezentarea tipică se face cu italice, la fel ca în cazul etichetei <DFN>.

<VAR> este asemănătoare cu <DFN> în privința modului de folosire; explică o variabilă sau un argument al unei bucăți de cod în contextul unei propoziții sau al unui paragraf normal. Figura 7.11 include un exemplu de folosire a etichetelor <CODE> și <VAR> la un loc. După listarea codului, sunt explicate pe scurt două variabile.

FIGURA 7.11. *Rândurile de cod Java sunt afișate cu un tip monospațiat pentru a le separa de textul explicativ.*



Alți doi camarazi ai elementelor de cod sunt folosiți în situații asemănătoare: <SAMP> și <KBD>.

Aceste două etichete identifică lucruri pe care calculatorul le comunică utilizatorului și exemple a ceea ce ar trebui utilizatorul să spună calculatorului. La fel ca etichetele <CODE> și <VAR>, ele există încă din primele zile ale Internetului, când aproape toată lumea lucra de pe un sistem UNIX bazat pe text. Sintaxa etichetei <SAMP> este următoarea:

```
<SAMP [core][international][events]>SampleOutput</SAMP>
```

SampleOutput este un exemplu de mesaj pe care utilizatorul l-ar putea primi de la calculator.

Sintaxa etichetei <KBD> este:

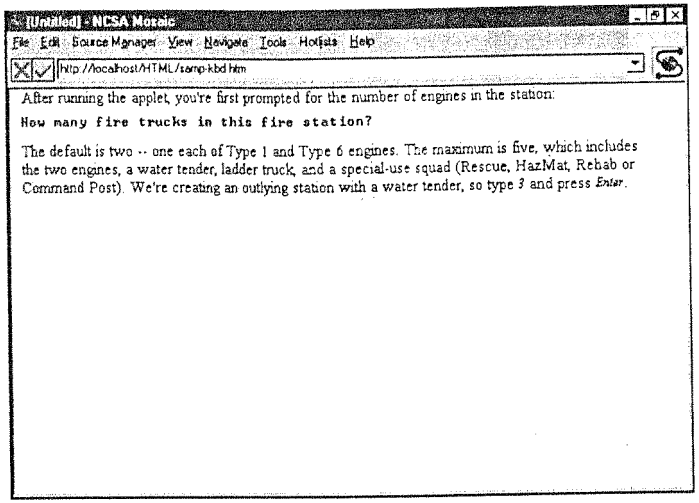
```
<KBD [core][international][events]>KeyboardInput</KBD>
```

KeyboardInput este un exemplu de ceea ce ar putea introduce utilizatorul de la tastatură. Reprezentarea obișnuită a eșantionului de ieșire este textul monospațiat, în timp ce pentru intrarea de la tastatură este folosit de obicei textul bold monospațiat, deși această reprezentare variază în funcție de browser. (Vezi Figura 7.12).

Alte formatări speciale ale textului

În domeniul formatării textului HTML, unele etichete nu se potrivesc bine într-o categorie sau alta. Următoarele stiluri nu sunt incluse printre elementele de frază sau de stil prezentate în Capitolul 17. Aceste etichete includ câteva formatări adiționale, bazate pe conținut.

FIGURA 7.12. *Ieșirea calculatorului este prezentată cu tipul bold monospațiat pe un rând separat; răspunsul utilizatorului este prezentat cu caractere italice.*



Documente care au suferit schimbări: <INS> și <DEL>

Una dintre noile adăugări la HTML 4 este abilitatea sa de a marca documentele pentru a arăta schimbările care s-au aplicat asupra lor. Această caracteristică a fost concepută pentru situațiile în care mai mulți oameni lucrează la același document. Când cineva introduce o bucată de text sau de conținut, are grijă să o plaseze între etichetele <INS>, iar conținutul ce trebuie șters, între etichetele <DEL>. Acest tip de marcare este des folosit de avocați când operează schimbări asupra unui document sau de autori și de redactori când revizuiesc o carte.

Când documentele trec prin mai multe mâini și revizuiri, acest mecanism este foarte folositor pentru că se poate vedea istoria revizuirii documentului.

Aceste două elemente sunt neobișnuite pentru HTML, deoarece ele nu sunt nici de nivel bloc, nici elemente inline. Ele pot să conțină unul sau mai multe cuvinte dintr-un paragraf sau să fie aplicate asupra unuia sau mai multor elemente de nivel bloc, cum ar fi paragrafele, listele și tabelele.

SUGESTIE

Etichetele <INS> și <DEL> nu sunt momentan acceptate de nici un browser, dar se așteaptă ca ele să apară în următoarele versiuni importante de Netscape sau Internet Explorer. O soluție posibilă o reprezintă textul subliniat pentru <INS> și tăiat cu o linie pentru <DEL>. Alte posibilități oferite de W3C sunt fonturile speciale, neafișarea textului șters sau alte marcări speciale.

Pentru momentul actual, ar fi o idee bună includerea unui stil pentru a reprezenta textul într-o manieră care să permită recunoașterea sa de către browserele care sunt măcar compatibile cu stilul.

Sintaxa pentru eticheta <INS> este următoarea:

```
<INS [cite=URL][datetime=date][core][international][events]>Content</INS>
```

Adresa *URL* indică un document sursă care arată de ce a fost schimbat materialul, iar *date* indică data și ora la care au fost făcute schimbările. Sintaxa este identică și pentru folosirea etichetei <DEL>.

SUGESTIE

Data și ora inserării trebuie introduse în conformitate cu standardul ISO8601, care dictează un format de forma YYYY-MM-DDThh:mm:ssTZD, după cum urmează:

- YYYY este un an reprezentat cu patru cifre.
- MM este luna reprezentată cu două cifre; 01 reprezintă ianuarie și așa mai departe.
- DD este ziua reprezentată cu două cifre.
- T este un separator care marchează începutul elementului de timp.
- hh reprezintă două cifre pentru oră, numai în modul de 24 de ore (de la 00 la 23).
- mm sunt două cifre pentru minute.
- ss sunt două cifre pentru secunde.
- TZD desemnează fusul orar, care este Z pentru Timpul Universal Coordonat (UTC) sau un număr de ore și minute (hh:mm) înaintea sau în urma UTC, reprezentat ca +hh:mm sau -hh:mm.

UTC a fost cunoscut mai demult ca Greenwich Mean Time (GMT), timpul local la meridianul Greenwich, aflat la zero grade longitudine. Inițialele pentru Timpul Universal Coordonat sunt preluate din traducerea în limba franceză a termenului.

Fiecare element *trebuie* să fie prezent pentru a reprezenta timpul. Ca exemplu, 27 august, 1997 5:42 p.m. în Mountain Standard Time este prezentat ca 1997-08-27T17:42:00-07:00.

Informația de adresă: <ADDRESS>

Eticheta <ADDRESS> este folosită pentru a marca informația de contact pentru documentul curent, fie că este o adresă e-mail sau o adresă completă de poștă și un număr de telefon. Se comportă în mare parte ca o etichetă de paragraf, forțând textul dintre limitele sale să se separe de materialul înconjurător prin spațierea adițională a rândului. Sintaxa arată în felul următor:

```
<ADDRESS [core][international][events]>...ContactInformation... </ADDRESS>
```

ContactInformation este informația de adresă, la un loc cu orice formatare de nivel paragraf, cum ar fi întreruperile de rând. De obicei este afișată ca un corp de text italic.

Puteți adăuga alte elemente în cadrul etichetei de adresă, inclusiv informații de hiperlegătură. Ca exemplu, iată o opțiune de a afișa informația de contact pentru o adresă e-mail:

```
<ADDRESS><A href="mailto:sam@fairdeals.com">Sam Beauregard</A><BR>
Fair Deal Sam's Used Cars<BR>
"With Sam B.,it's a guarantee"<BR>
Great Falls, MT
</ADDRESS>
```

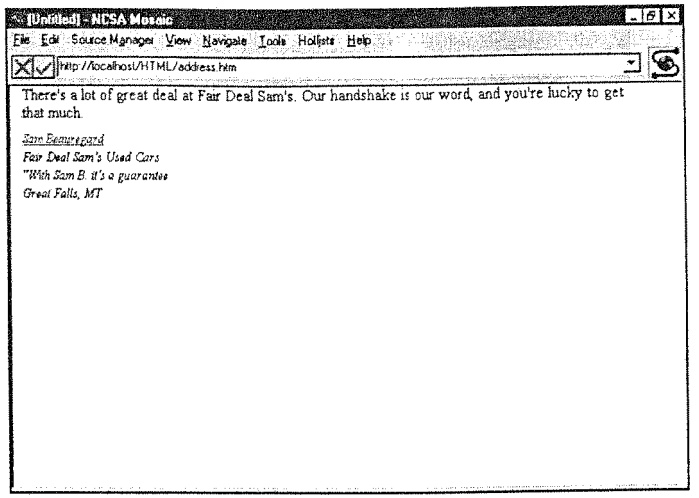
Acest cod o să creeze o hiperlegătură pentru a se conecta la programul de e-mail al utilizatorului în timp ce se menține formatul pentru informațiile despre adresă. (Vezi Figura 7.13). Pentru mai multe idei despre cum se lucrează cu hiperlegături și destinații mailto, vezi Capitolul 13.

Lucrul cu citate lungi și scurte: <BLOCKQUOTE> și <Q>

Pentru citate mai scurte, este bine să folosiți ghilimele sau eticheta <Q> și să lăsați textul în același rând cu restul conținutului. Totuși, dacă mărimea textului citat depășește câteva propoziții, este mai ușor pentru cititor să observe citatul dacă acesta este separat de restul textului. În acest caz, este folosit elementul <BLOCKQUOTE>. Sintaxa este:

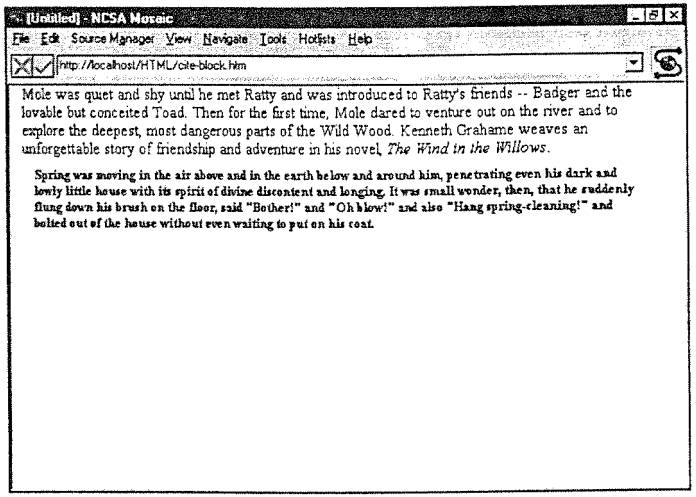
```
<BLOCKQUOTE [cite=URL][core][international][events]>...Text...</BLOCKQUOTE>
```


FIGURA 7.13. *Text formatat cu eticheta <ADDRESS>; apare o etichetă imbricată pentru definirea unei hiperlegături.*



Text este citatul care trebuie separat de restul materialului înconjurător, iar *URL* indică sursa citatului. Eticheta <BLOCKQUOTE> creează un paragraf separat pentru text și, în majoritatea browserelor, indentează întregul paragraf de la stânga. (Vezi Figura 7.14). Unele browsere includ de asemenea o ușoară indentare de la dreapta sau reprezentarea citatului cu caractere italice.

FIGURA 7.14. *Un citat bloc; sursa citatului este marcată prin caractere italice.*



Un citat bloc permite altă formatare între granițele sale, inclusiv paragrafe, întreruperi de rând și titluri. Fiecare browser manevrează diferit formatările speciale. Majoritatea vor indenta formatările speciale, cum ar fi titlurile, la un loc cu restul citatului bloc.

Rețineți că nu toate browserele tratează citatele bloc în același fel, în special din punct de vedere al stilului caracterelor (normale sau italice). Dacă includeți formatări cum ar fi caracterele italice în cadrul unui citat bloc, iar browserul va afișa blocul cu italice, atunci

evidențierea dumneavoastră nu ajunge la cititor. În unele cazuri, browserul ar putea schimba stilul caracterelor în restul citatului bloc.

SUGESTIE

Pentru siguranță și aspect unitar, încercați să evitați folosirea altor formatări în cadrul blocului citat.

Eticheta <Q> este destinată folosirii pentru citate mai scurte care nu necesită o deplasare față de restul textului, așa cum se face în cazul etichetei BLOCKQUOTE. Citatul bloc este un element de bloc care include elemente de paragraf înainte și după, dar citatul Q este un element inline. Sintaxa sa este următoarea:

```
<Q [cite=URL][core][international][events]>...Text...</Q>
```

Text este citatul care ar trebui separat de restul materialului înconjurător, iar *URL* indică sursa citatului. Aceasta este o etichetă nouă, care nu este încă implementată de majoritatea browserelor.

Desenarea unei linii pe pagină: <HR>

Liniile orizontale sunt un element ușor de folosit pentru a diviza o pagină în secțiuni logice. Ele avertizează cititorul să fie atent la o schimbare a subiectului sau al stilului sau pot separa figuri și denumiri de corpul de text. În funcție de necesitățile designului dumneavoastră, <HR> include câteva atribute pentru un control mai fin al înfățișării. Sintaxa este:

```
<HR [align][width=lineWidth][size=lineThicknes][noshade][core][events]>
```

În acest caz, *lineWidth* și *lineThicknes* setează aspectul de bază al acestei linii. Înălțimea unei linii orizontale este fixată în pixeli, cum ar fi size=8. Lățimea este fixată fie în pixeli, fie ca procent din lățimea ferestrei browserului. Dacă folosiți valoarea procentuală, includeți un semn de procent (%) după ea. Standardul prestabilit este de 3 pixeli înălțime și 100 procente din pagină lățime. (Vezi Figura 7.15).

Alinierea prestabilită a unei linii orizontale este de obicei de la stânga. Liniile nu pot fi aliniate stânga-dreapta – încercați să setați lățimea la 100%.

Ultimul atribut este noshade. În mod normal, un browser afișează o linie orizontală de separare într-o formă de umbrire tridimensională. Această umbrire variază de la browser la browser; unele o arată ca fiind adâncită, unele înălțată, unele ca o linie de contur. Folosind noshade, forțăm linia orizontală să apară ca o linie groasă fără nici o altă evidențiere. Din nou, această înfățișare poate varia la browsere. Unele folosesc negrul pentru linie, altele folosesc nuanțe de gri.

O rapidă trecere în revistă a stilurilor logice

Poate ați observat multe suprapuneri în felul în care elemente logice diferite din HTML sunt reprezentate într-un browser. Deși acestea nu sunt setate standard, există unele convenții acceptate de toată lumea. Tabelul 7.1 prezintă stilurile logice și redarea obișnuită.

Tabelul 7.1. Stilurile elementelor logice.

Element	*Stil*
<ADDRESS>	*Italic*
<BLOCKQUOTE>	Normal cu indentare
<CITE>	*Italic*

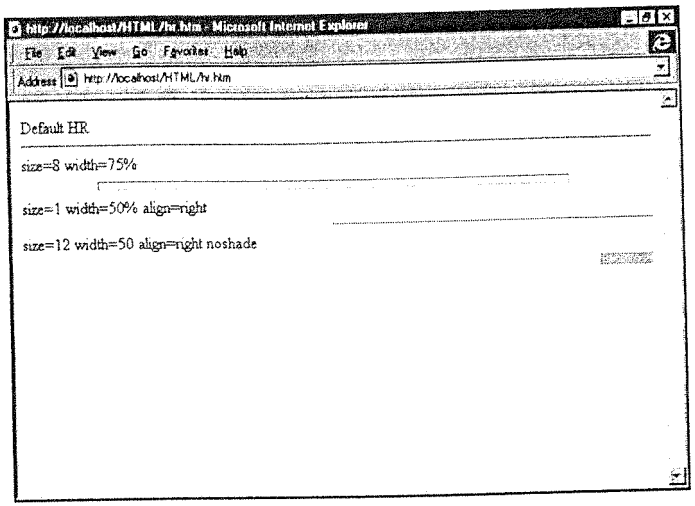


FIGURA 7.15. *Prima linie din această pagină folosește valorile prestabilite ale browserului. La rândurile următoare valorile* hight *(înălțime) și* width *(lățime) sunt ajustate pentru diferite efecte.*

Element	*Stil*
<CODE>	Monospace
<DFN>	*Italic*
<EM>	*Italic*
<KBD>	Bold Monospace
<SAMP>	Monospace
<STRONG>	Bold
<VAR>	*Italic*

Browserele le pot interpreta pe acestea într-o multitudine de feluri, dar țineți minte că aceasta nu deranjează. O parte din scopul HTML-ului este crearea unor documente independente de platformă. Utilizatorului nu ar trebui să-i pese care stil este folosit pentru fiecare din elemente, câtă vreme este consecvent.

Când folosiți stiluri logice, browserul poate decide cum să afișeze un anumit tip de text bazat pe capacitățile platformei, reducându-vă grijile referitoare la locul unde poate fi văzut documentul dumneavoastră.

Rezumat

Scopul acestui capitol este să vă ofere un start lansat în formatarea textului pe paginile dumneavoastră Web, astfel încât mesajul pe care îl transmiteți să aibă un impact maxim. Cu excepția câtorva etichete pentru hiperlegături și imagini, acest capitol include toate etichetele folosite cel mai frecvent în HTML.

În primul rând, și cele mai importante, sunt etichetele care despart textul în paragrafe și rânduri. Fără ele, un document HTML ar fi doar un text lung și greu de citit. Ele furnizează structura de bază a paginii pentru cititor.

După spargerea documentului în paragrafe, HTML 4 oferă un set de etichete care au un sens logic și structural, în defavoarea informațiilor explicite despre înfățișare, pentru

formatarea paginii. Aceste etichete formatează textul după scopul și folosirea sa, cum ar fi citatele și adresele. Ele sunt de obicei acceptate pe toate platformele și browserele, chiar dacă redarea reală poate fi atipică.

O multitudine de alte etichete HTML controlează de asemenea înfățișarea și formatarea textului, dar particularitățile lor impun o prezentare individuală. Printre acestea se numără etichetele de liste (Capitolul 8, „Folosirea listelor pentru organizarea informațiilor"), etichete de stil (Capitolul 17, „Stil simplu cu text"), hiperlegături și ancore (Capitolul 13, „Hiperlegături și ancore"), tabele și cadre (Capitolele 10, „Macheta și prezentarea tabelelor" și 20, „Crearea cadrelor") și formulare (Capitolul 15, "Crearea și utilizarea formularelor HTML").

Capitolul 8
Folosirea listelor pentru organizarea informațiilor

BIBL. CENTR. UNIV. „MIHAIL EMINESCU" IAȘI SECȚIA INFORMATICĂ

În acest capitol

Liste ordonate (sau numerotate): <OL> 99
Liste neordonate: <UL> 108
O listă de definiții sau de glosar: <DL> 111
Folosirea stilurilor de listă DIR și MENU 113

Listele sunt pretutindeni. Pe un frigider pentru cumpărături, pe un carnet de însemnări pentru lucruri ce trebuie făcute, la începutul cărților sub forma unui cuprins, în materialele transmise de un șef pentru executarea unor sarcini. Listele sunt unul dintre cele mai normale moduri de a organiza informațiile.

HTML are un set special de etichete pentru afișarea listelor, împreună cu o mulțime de atribute specifice, pentru a vă furniza un control mai bun asupra înfățișării acestora.

La nivelul cel mai de jos, listele se împart în două categorii:

- **Liste ordonate:** Aceste liste sunt de obicei folosite pentru a indica o succesiune de evenimente sau proprietăți. Mai sunt folosite și pentru a identifica exact secțiuni și relații în momentul creării structurilor.
- **Liste neordonate:** O listă neordonată este în mod obișnuit folosită pentru a afișa un grup de obiecte oarecum legate între ele, dar nu neapărat prezentate în manieră ierarhică. Trei subseturi HTML speciale de liste neordonate sunt ilustrate mai târziu în acest capitol – definiții, directoare și meniuri.

Următoarele secțiuni discută fiecare tip de listă și metodele de a le personaliza pentru situații speciale.

Liste ordonate (sau numerotate): <OL>

O listă este definită prin etichetele sale de deschidere și de închidere. Pentru liste ordonate, acestea sunt `<OL>` și `</OL>`. Totuși, nu vă vor fi suficiente etichetele de deschidere și de închidere – veți avea nevoie să puneți ceva între ele. Priviți Listingul 8.1, care este o listă închisă de etichetele de listă ordonată.

Listingul 8.1. O listă ordonată simplă, specificată cu etichetele <OL>.

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0//EN">
<HTML>
<HEAD>
<TITLE>Basic CPR</TITLE>
</HEAD>
<BODY>
Basic CPR
<OL>
```

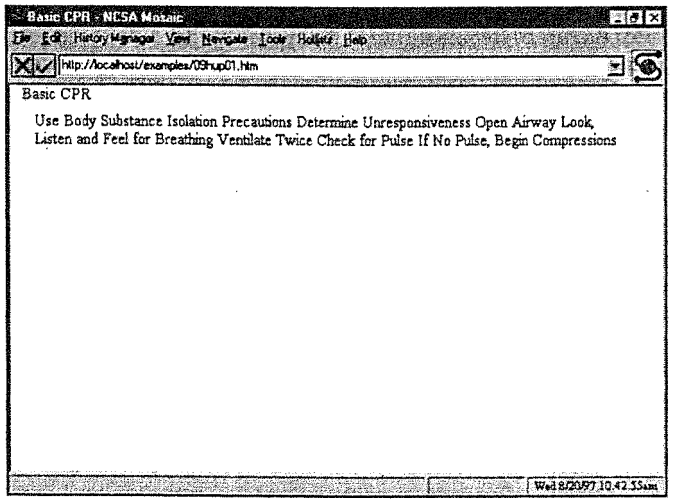
(continuare)

Listingul 8.1. Continuare

```
Use Body Substance Isolation Precautions
Determine Unresponsiveness
Open Airway
Look, Listen and Feel for Breathing
Ventilate Twice
Check for Pulse
If No Pulse, Begin Compressions
</OL>
</BODY>
</HTML>
```

Dacă lista este văzută pe un browser, vă veți da seama că textul este conținutul unei liste, dar nu apare nici o numerotare. (Vezi Figura 8.1).

FIGURA 8.1.
O listă de informații închisă de etichetele de listă nu va rezulta într-o listă HTML.



Cum se face că lista nu are numere? Amintiți-vă că HTML nu recunoaște întreruperile de rând decât dacă îi este comunicat în mod special unde își au acestea locul de aplicare prin etichetele <P> sau <BR>. Același principiu se aplică listei, dar este folosită o etichetă diferită pentru a spune browserului unde începe fiecare articol în parte. Aceasta este eticheta articolului de listă, <LI>.

Cu etichetele <OL> și <LI> la îndemână, sintaxa de bază pentru o listă ordonată este:

```
<OL>
<LI>ListItem1
<LI>ListItem2
...
<LI>ListItemN
</OL>
```

ListItem este fiecare articol în parte din listă. În Listingul 8.2 se arată aceeași listă folosită în Listingul 8.1, ca un exemplu despre cum să folosiți aceste etichete.

OBSERVAȚIE

Pot fi folosite atributele standard pentru elementele de listă ordonată și articolele listei:

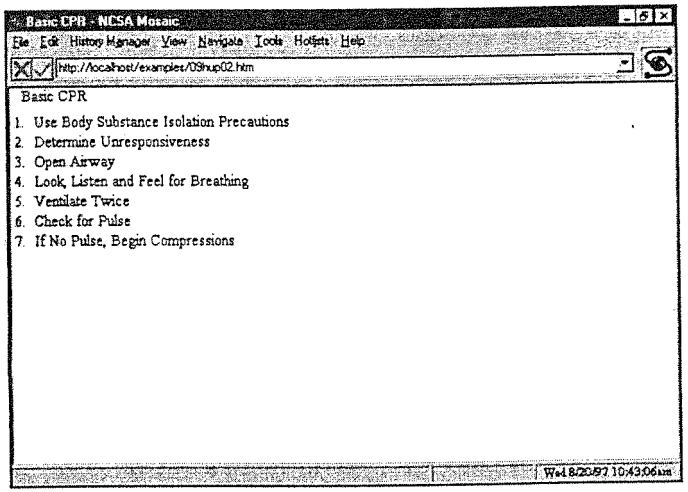
- `id, class, style, title` (Capitolul 6, „Structurarea textului")
- `lang, dir` (Capitolul 40, „Internaționalizarea limbajului HTML")
- `onclick, ondblclick, onmousedown, onmouseup, onmouseover, onmousemove, onmouseout, onkeypress, onkeydown, onkeyup` (Capitolul 28, „Lucrul cu utilizatorul – Evenimente")

Listingul 8.2. Marcarea începutului fiecărui articol cu eticheta articolului de listă.

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0//EN">
<HTML>
<HEAD>
<TITLE>Basic CPR</TITLE>
</HEAD>
<BODY>
Basic CPR
<OL>
<LI>Use Body Substance Isolation Precautions
<LI>Determine Unresponsiveness
<LI>Open Airway
<LI>Look, Listen and Feel for Breathing
<LI>Ventilate Twice
<LI>Check for Pulse
<LI>If No Pulse, Begin Compressions
</OL>
</BODY>
</HTML>
```

Pentru fiecare articol identificat cu <LI>, browserul începe un rând nou, îl indentează și adaugă un număr. (Vezi Figura 8.2).

FIGURA 8.2.
După ce fiecare articol este marcat cu o etichetă de articol de listă, browserul îl indentează și apoi adaugă automat numerotarea.



SUGESTIE

Puteți include, de asemenea, articlolelor din listă și o etichetă opțională de închidere de listă, </LI>. Nu este obligatorie deoarece începutul articolului următor din listă implică sfârșitul articolului precedent.

Totuși, dacă aveți de gând să folosiți în listă foile de stil (vezi Partea a IV-a, „Tehnici de prezentare") sau caracteristicile Dynamic HTML (vezi Partea a VI-a, „HTML dinamic"), eticheta de închidere este necesară.

Sunt permise și etichetele de întrerupere de rând, <P> și <BR>, în cadrul corpului unei liste, pentru a controla mai bine înfățișarea și formatarea sa, după cum se vede în Listingul 8.3.

**Listingul 8.3. Organizarea listei dumneavoastră pe titluri și subtexte cu etichetele
 și <P>.**

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0//EN">
<HTML>
<HEAD>
<TITLE>Basic CPR</TITLE>
</HEAD>
<BODY>
Basic CPR
<OL>
<LI>Use Body Substance Isolation Precautions<BR>
If available, be sure to use latex or vinyl gloves,
goggles and a barrier device with one-way valve.<P>
<LI>Determine Unresponsiveness
<LI>Open Airway
<LI>Look, Listen and Feel for Breathing
<LI>Ventilate Twice
<LI>Check for Pulse
<LI>If No Pulse, Begin Compressions
</OL>
</BODY>
</HTML>
```

Când sunt văzute pe un browser (după cum se observă în Figura 8.3), etichetele care afectează întreruperile de rând nu afectează numerotarea rândurilor. Când lista începe cu eticheta <OL>, numai o etichetă <LI> poate face browserul să înceapă un rând nou cu numărul următor. Puteți forța și alte rânduri să înceapă cu etichetele de întrerupere de rând, dar, deoarece nu începe cu un nou articol al listei, browserul îl tratează ca pe o continuare a articolului curent și nu adaugă un număr pentru următorul articol din secvență.

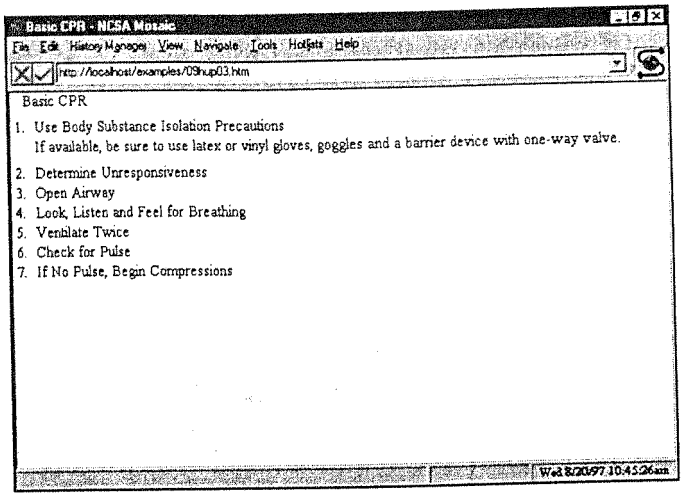
OBSERVAȚIE

Pot fi folosite elementele standard de formatare în cazul elementelor <LI>:

- **elemente de font:** TT, I, B, U, S, STRIKE, BIG, SMALL (Capitolul 17, "Stil simplu cu text")
- **elemente de frază:** EM, STRONG, DFN, CODE, SAMP, KBD, VAR, CITE, ACRONYM (Capitolul 7, „Alinierea și formatarea textului")

Pentru un cod mai clar și revizuirile ulterioare, asigurați-vă că orice etichetă de formatare deschisă într-un articol din listă este închisă, înainte ca următorul articol să înceapă.

FIGURA 8.3.
Formatarea adițională a textului și întreruperile de pagină nu afectează numerotarea și stilul de bază al listei.



Pe lângă formatarea articolelor individuale de listă, mai există două metode de a personaliza o listă ordonată: alegerea unui nou început pentru numerotarea listei (start) și selectarea caracterelor de numerotare folosite (type). Aceste două atribute sunt discutate în secțiunile următoare.

Unde să începeți o listă

Câteodată este necesară întreruperea listei pentru un text explicativ sau alt material fără a fi necesară o indentare. Priviți Listingul 8.4, care arată două liste ordonate întrerupte de un text explicativ și căruia îi corespunde Figura 8.4.

Listingul 8.4. Întreruperea unei liste ordonate cu un text explicativ pe care nu vreți să-l indentați.

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0//EN">
<HTML>
<HEAD>
<TITLE>Basic CPR</TITLE>
</HEAD>
<BODY>
<H2>Basic CPR</H2>
<H3>Airway and Breathing</H3>
<OL>
<LI>Use Body Substance Isolation Precautions<BR>
If available, be sure to use latex or vinyl gloves,
goggles and a barrier device with one-way valve.<P>
<LI>Determine Unresponsiveness
<LI>Open Airway
<LI>Look, Listen and Feel for Breathing
<LI>Ventilate Twice
</OL>
<P>Look for chest rise and listen for exhalation while ventilating.
If no air enters, reposition the airway and try again. If the
airway is blocked, attempt to clear it using appropriate airway
obstruction maneuvers. If you can't get air into the patient,
```

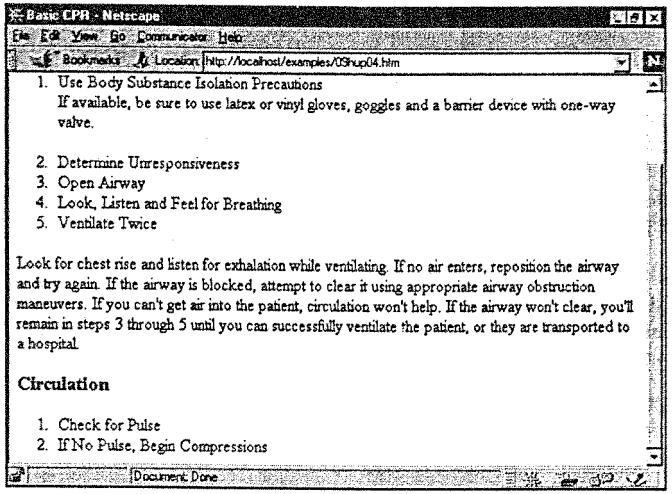
(continuare)

Listingul 8.4. Continuare

```
circulation won't help. If the airway won't clear, you'll remain
in steps 3 through 5 until you can successfully ventilate the
patient, or they are transported to a hospital.</P>
<H3>Circulation</H3>
<OL>
<LI>Check for Pulse
<LI>If No Pulse, Begin Compressions
</OL>
</BODY>
</HTML>
```

În Figura 8.4, numerotarea nu este consecutivă pentru listele individuale. Procesul de numerotare este luat de la început pentru fiecare listă.

FIGURA 8.4.
Fără atributul `start`*, numerotarea nu este consecutivă între liste.*



Sunt folosite două seturi de etichete `<OL>` – una pentru prima jumătate din CPR (pasaje respiratorii și respirație) și alta pentru a doua jumătate (circulație). Deoarece fiecare porțiune a listei este încadrată de propria sa pereche de etichete de listă ordonată, browserul tratează fiecare porțiune ca pe o listă separată și începe numerotarea de la 1 în cea următoare. Browserul nu știe că aceasta este de fapt o continuare a primei liste.

Pentru a ocoli această problemă, folosiți eticheta `<OL>` cu atributul `start`, pentru a începe numerotarea de oriunde vreți. Acest atribut poate fi folosit pentru a anula numerotarea prestabilită a listei când aceasta are mai multe părți. (Vezi Listingul 8.5 și Figura 8.5.)

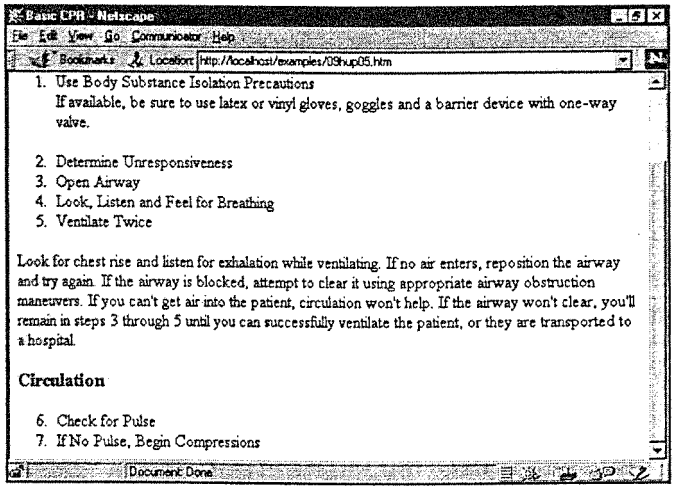
Listingul 8.5. Folosirea atributului `start`.

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0//EN">
<HTML>
<HEAD>
<TITLE>Basic CPR</TITLE>
</HEAD>
<BODY>
```



```
<H2>Basic CPR</H2>
<H3>Airway and Breathing</H3>
<OL>
<LI>Use Body Substance Isolation Precautions<BR>
If available, be sure to use latex or vinyl gloves, goggles and a barrier device
with one-way valve.<P>
<LI>Determine Unresponsiveness
<LI>Open Airway
<LI>Look, Listen and Feel for Breathing
<LI>Ventilate Twice
</OL>
<P>Look for chest rise and listen for exhalation while ventilating.
If no air enters, reposition the airway and try again. If the
airway is blocked, attempt to clear it using appropriate airway
obstruction maneuvers. If you can't get air into the patient,
circulation won't help. If the airway won't clear, you'll remain
in steps 3 through 5 until you can successfully ventilate the
patient, or they are transported to a hospital.</P>
<H3>Circulation</H3>
<OL start=6>
<LI>Check for Pulse
<LI>If No Pulse, Begin Compressions
</OL>
</BODY>
</HTML> parts
```

FIGURA 8.5.
Folosirea atributului `start` *cu eticheta* `<OL>`.



Folosirea atributului `start` face posibilă crearea unor liste care sunt întrerupte de blocuri de text sau de alt material și apoi reluate cu numerotarea potrivită. Aceasta este sintaxa pentru `start`:

```
<OL start=number>
```

În acest atribut, *number* este un număr întreg între 2147483647 și –2147483648. Nu aveți voie să folosiți virgule în numerele dumneavoastră, deoarece acestea vor fi interpretate greșit de către browser.

SUGESTIE

Dacă sunt folosite numere din afara domeniului, acestea o vor lua de la început. Ca exemplu, dacă o listă începe cu 2147483647, numărul următor afișat în rând va fi –2147483648. Dacă listele dumneavoastră sunt cotate de un sistem care se întinde dincolo de două miliarde și ceva în oricare dintre direcții, probabil că aveți prea mult timp liber.

De ce tip de listă este nevoie?

În primele zile ale HTML-ului, exista doar un tip de listă ordonată, cu numere începând de la 1 și până unde se oprea lista. Apoi, pe măsură ce oamenii au început să extindă folosirea listelor către lucruri cum ar fi cărțile online, au vrut să folosească și altceva în afară de numere. Pentru a satisface această cerință a fost creat atributul `type`; sintaxa sa este:

```
<OL type=numberingSystem>
```

În acest atribut, *`numberingSystem`* este unul dintre următoarele cinci caractere: `1`, `A`, `a`, `I` sau `i`. Exemple pentru unitățile de numerotare sunt ilustrate în Tabelul 8.1.

Tabelul 8.1. Valori și stiluri ale atributului `type`.

Valoare	*Stil*	*Exemplu*
`1`	Arabic	1, 2, 3...
`A`	Litere mari	A, B, C...
`a`	Litere mici	a, b, c...
`I`	Cifre romane mari	I, II, III...
`i`	Cifre romane mici	i, ii, iii...

Posibilitatea de a folosi altceva în afară de numere a condus către o caracteristică folositoare a listelor ordonate: *listele imbricate*. Acestea sunt liste în cadrul altor liste și se pot întinde pe mai multe niveluri. Pentru a crea o listă imbricată, includeți un nou set de etichete de listă ordonată în cadrul etichetelor actuale ale listei (după cum se vede în Listingul 8.6). Browserul începe o nouă listă pentru etichetele noi, dar ține cont unde rămăsese listă inițială după ce se termină cea imbricată. (Vezi Figura 8.6.)

Listingul 8.6. Crearea unei liste imbricate.

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0//EN">
<HTML>
<HEAD>
<TITLE>Patient Assessment</TITLE>
</HEAD>
<BODY>
<!--BEGIN MAIN LIST-->
<OL type=A>

<LI>Safety Considerations
<OL type=1>
  <LI>Body substance isolation
  <LI>Scene safety
  <LI>Initial size-up
</OL>

<LI>Initial Patient Assessment
```



```
<OL type=1>
  <LI>General Impression
  <LI>Unresponsiveness
  <OL type=i>
    <LI>Alert to person, place and time
    <LI>Verbal response to audible stimuli
    <LI>Pain evokes verbal or physical response
    <LI>Unresponsiveness to all stimuli
  </OL>
</OL>

<LI>Patient Critical Needs
<OL type=1>
  <LI>Airway
  <LI>Breathing
  <OL type=i>
    <LI>Use oxygen if indicated
    <LI>Consider use of assisting with bag valve mask
  </OL>
  <LI>Circulation
  <LI>Bleeding
</OL>
<!--END MAIN LIST-->
</OL>
</BODY>
</HTML>
```

Una dintre principalele probleme la imbricarea listelor este confuzia pe care o pot genera. Este ușor să uităm care listă e care și care este subordonată căreia după al doilea sau al treilea set înglobat de articole. Țineți minte că utilizarea caracterelor Carriage Return nu introduce interliniere și spații posterioare suplimentare. Folosiți tasta ENTER și indentări cât mai dese în etichetele de comentariu pentru a face codul sursă al paginii dumneavoastră să fie mai ușor de citit și de editat.

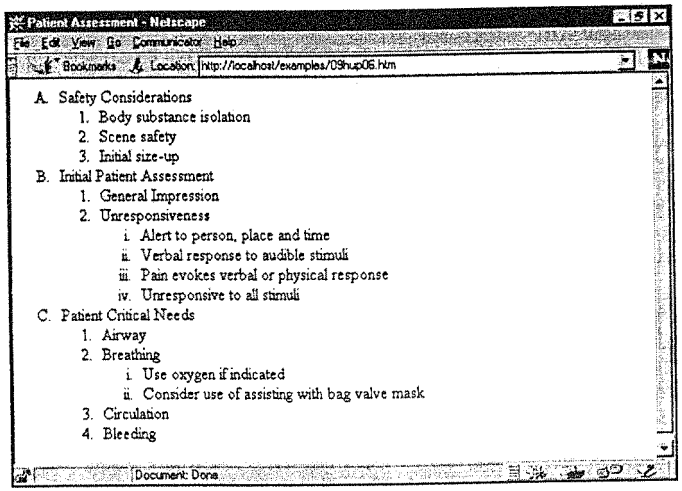


FIGURA 8.6. *Folosirea atributului* `type` *pentru listele imbricate conduce la formatarea textului într-un formular structurat.*

SUGESTIE

Dacă folosiți `type` cu `start`, folosiți un număr întreg pentru `start`, indiferent de ce tip este specificat prin `type`.

De exemplu, dacă includeți o listă ordonată care folosește litere mari și începe de la litera *D*, eticheta de deschidere ar trebui să fie `<OL type=A start=4>`. Browserul va traduce poziția de început în litera corespunzătoare.

Acum, că ați lucrat cu diferite tipuri și variațiuni de listă ordonată, este timpul să privim spre sora sa vitregă – lista neordonată.

Liste neordonate: <UL>

Listele neordonate reprezintă de obicei un set de articole care sunt legate unul de celălalt într-o oarecare măsură, dar nu trebuie neapărat să respecte o anumită ordine. Sintaxa este asemănătoare cu cea a unei liste ordonate:

```
<UL>
<LI>ListItem1
<LI>ListItem2
...
<LI>ListItemN
</UL>
```

ListItem este fiecare aricol în parte. Vezi Listingul 8.7 pentru exemplificare.

OBSERVAȚIE

Pot fi folosite atributele standard pentru elementele listei neordonate și articolele listei:

- `id`, `class`, `style`, `title` (Capitolul 6)
- `lang`, `dir` (Capitolul 40)
- `onclick`, `ondblclick`, `onmousedown`, `onmouseup`, `onmouseover`, `onmousemove`, `onmouseout`, `onkeypress`, `onkeydown`, `onkeyup` (Capitolul 28)

OBSERVAȚIE

Pot fi folosite elementele standard de formatare în cadrul elementelor listelor neordonate `<LI>`:

- **elemente de font:** `TT`, `I`, `B`, `U`, `S`, `STRIKE`, `BIG`, `SMALL` (Capitolul 17, "Stil simplu cu text")
- **elemente de frază:** `EM`, `STRONG`, `DFN`, `CODE`, `SAMP`, `KBD`, `VAR`, `CITE`, `ACRONYM` (Capitolul 7, „Alinierea și formatarea textului")

Pentru un cod mai clar și revizuirile ulterioare, asigurați-vă că orice etichetă de formatare deschisă într-un articol din listă este închisă, înainte ca următorul articol să înceapă.

Listingul 8.7. Substituirea lui <OL> cu <UL> într-o listă neordonată.

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0//EN">
<HTML>
<HEAD>
<TITLE>Jump Kit Inventory</TITLE>
</HEAD>
<BODY>
EMS Jump Kit Contents
<UL>
<LI>Rescue Scissors and Penlight
<LI>Stethoscope and Sphygmanometer
<LI>Oxygen Bottle
<LI>Non-Rebreather Mask and Nasal Cannula
<LI>Oral and Nasal Airways
<LI>Gauze and Trauma Dressings
<LI>Sterile Saline and Water
<LI>Oral Glucose and Activated Charcoal
</UL>
</BODY>
</HTML>
```


După cum puteți vedea în Figura 8.7, lista neordonată este de obicei reprezentată în același fel ca o listă ordonată, cu excepția că în loc de numere sunt marcaje de listă.

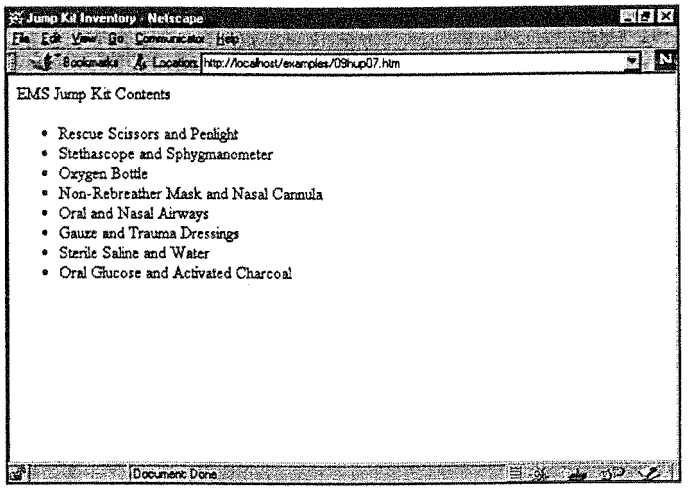


FIGURA 8.7. *O listă neordonată folosește marcaje de listă în loc de numere.*

Înfățișarea reală a unui marcaj de listă variază în funcție de browser. Internet Explorer folosește marcaje mici tip text, iar NCSA Mosaic folosește marcaje grafice.

Evident că numerotarea nu este o opțiune pentru listele neordonate. Totuși, HTML 4 oferă unele opțiuni adiționale pentru prestabilirea tipului de marcaj de listă cu atributul `type`.

Ce tip de marcaj de listă doriți?

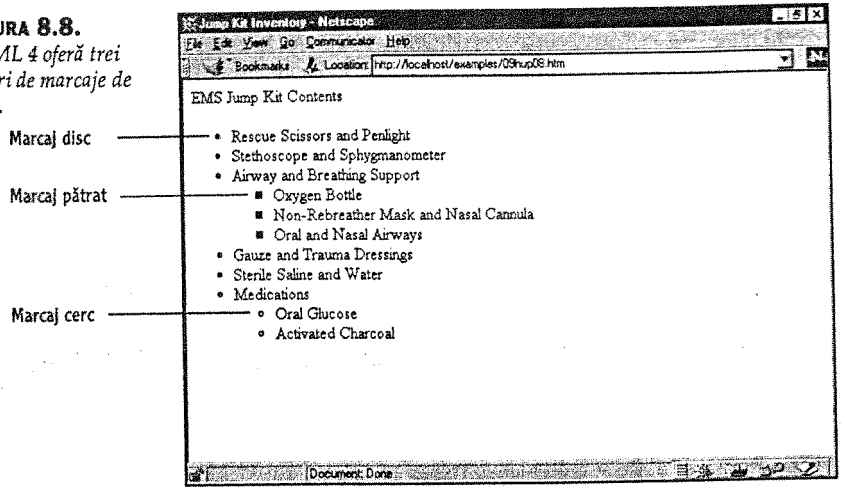
HTML 4 oferă posibilitatea de a utiliza trei tipuri de bază de marcaje de listă, deși nu toate browserele le acceptă pe toate trei. Dacă un browser nu recunoaște atributul, folosește marcajul prestabilit. Sintaxa pentru `type` este:

```
<UL type=bulletType>
```

În acest atribut, *bulletType* poate avea una dintre următoarele trei valori: `circle`, `square` sau `disc`. Reprezentările lor sunt arătate în Figura 8.8.

De asemenea, la fel ca în listele ordonate, o listă neordonată poate conține o listă în cadrul alteia. Imbricarea ajută la arătarea dependenței dintre articole, deși nu este stabilită o ierarhie. Această caracteristică este implementată în același fel, cu alt set de etichete de listă înlocuind un articol al listei. (Vezi Listingul 8.8.) Atributul `type` ajută la diferențierea grafică a dependențelor.

FIGURA 8.8.
HTML 4 oferă trei tipuri de marcaje de listă.



Listingul 8.8. Folosirea diferitelor tipuri de marcaje de listă ajută la diferențierea grafică a sublistelor.

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0//EN">
<HTML>
<HEAD>
<TITLE>Jump Kit Inventory</TITLE>
</HEAD>
<BODY>
EMS Jump Kit Contents
<UL type=disc>
<LI>Rescue Scissors and Penlight
<LI>Stethoscope and Sphygmanometer

<LI>Airway and Breathing Support
<UL type=square>
<LI>Oxygen Bottle
<LI>Non-Rebreather Mask and Nasal Cannula
<LI>Oral and Nasal Airways
</UL>

<LI>Gauze and Trauma Dressings
<LI>Sterile Saline and Water

<LI>Medications
<UL type=circle>
<LI>Oral Glucose
<LI>Activated Charcoal
</UL>
</UL>
</BODY>
</HTML>
```

Observați că articolul de listă de dinaintea listei înglobate este de asemenea un titlu pentru listă. Acest articol ajută cititorul, deoarece el identifică modul în care sublista se leagă de lista principală.

OBSERVAȚIE

Mai este permis și un alt atribut pentru cele două tipuri de listă, deși implementarea sa în browsere nu este prea bună și este acum nerecomandat de HTML 4. Acest atribut este `compact`, care cere clientului să încerce afișarea listei în modul cel mai eficient posibil din punct de vedere al spațiului. Având în vedere celelalte variabile posibile dintr-o listă, majoritatea browserelor ignoră acest atribut.

O listă de definiții sau de glosar: <DL>

Ultima etichetă specială pentru liste este `<DL>`, care reprezintă o listă de definiții. Acestă etichetă este folosită la crearea unei listări tip glosar, care este ușor de folosit pentru articole cum ar fi dicționarele sau paginile FAQ.

Eticheta `<DL>` este folosită asemănător cu eticheta de listă neordonată, cu excepția că nu folosește eticheta `<LI>` pentru a marca intrările sale, deoarece o listă de definiții necesită două articole pentru fiecare intrare: un termen și definiția sa. Marcarea acestor două articole este realizată cu etichetele corespondente `<DT>` și `<DD>`. Sintaxa pentru eticheta `<DL>` este următoarea:

```
<DL>
<DT>Term1<DD>Definition1
<DT>Term2<DD>Definition2
...
<DT>TermN<DD>DefinitionN
</DL>
```

În această etichetă, *`Term`* este cuvântul care trebuie definit, iar *`Definition`* este blocul de text care furnizează definiția. Altă formă de a include etichetele pe pagină plasează fiecare etichetă `<DT>` și `<DD>` pe rânduri diferite, dar în prima metodă scopul este puțin mai clar. Browserul prezintă conținutul în aceeași manieră (pe rânduri separate), indiferent de metoda pe care o alegeți.

OBSERVAȚIE

Pot fi folosite atributele standard pentru elementele listei de definiții:

- `id, class, style, title` (Capitolul 6)
- `lang, dir` (Capitolul 40)
- `onclick, ondblclick, onmousedown, onmouseup, onmouseover, onmousemove, onmouseout, onkeypress, onkeydown, onkeyup` (Capitolul 28)

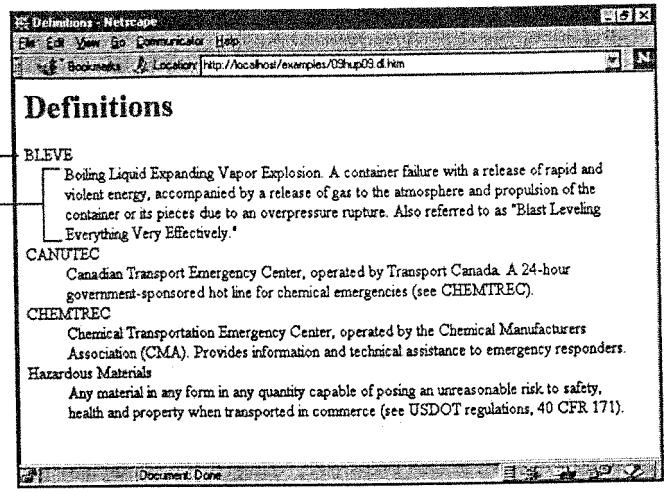
Implementarea etichetei `<DL>` poate varia un pic în funcție de browser. În Figura 8.9 este prezentată o metodă tipică de afișare a conținutului.

Ca și la celelalte etichete și stiluri de liste, este posibilă includerea etichetelor de formatări fizice adiționale în termeni sau în definiții:

```
<DT><B>Term</B>
<DD><I>Definition</I>
```

O metodă des folosită pentru implementarea listei de definiții constă în a da termenului un stil bold sau italic pentru definiții (după cum se vede în Figura 8.10). Nici eticheta `<DT>` și nici eticheta `<DD>` nu include atribute proprii, altele decât opțiunile standard (vezi observația precedentă).

FIGURA 8.9. *Folosirea unei serii de indentări agățate pentru a separa termenii de definiții.*



OBSERVAȚIE

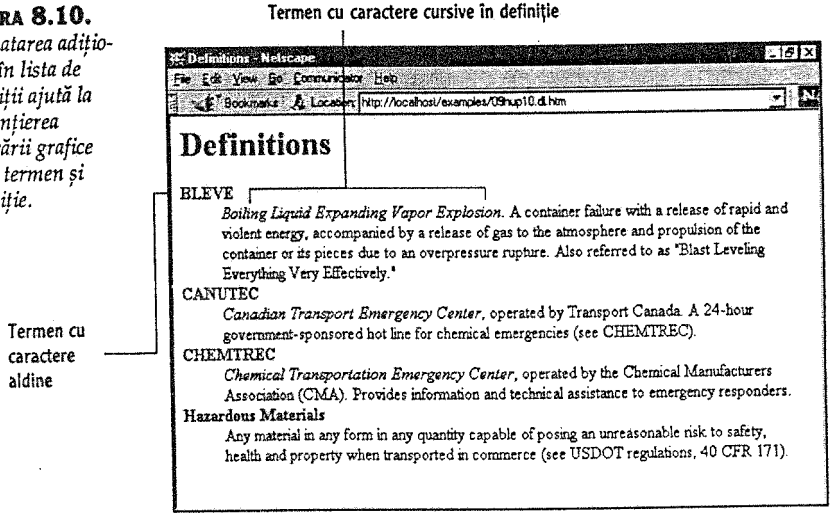
Pot fi folosite elementele standard de formatare în cadrul elementelor `<DT>` și `<DD>`:

- **elemente de font:** `TT`, `I`, `B`, `U`, `S`, `STRIKE`, `BIG`, `SMALL` (Capitolul 17)
- **elemente de frază:** `EM`, `STRONG`, `DFN`, `CODE`, `SAMP`, `KBD`, `VAR`, `CITE`, `ACRONYM` (Capitolul 7)

Pentru un cod mai clar și revizuirile ulterioare, asigurați-vă că orice etichetă de formatare deschisă într-un articol din listă este închisă, înainte ca următorul articol să înceapă.

Acum, că ați văzut stilurile principale pentru crearea și afișarea listelor, a venit timpul să privim și ultimele două stiluri.

FIGURA 8.10. *Formatarea adițională în lista de definiții ajută la evidențierea separării grafice între termen și definiție.*



Folosirea stilurilor de listă DIR și MENU

Ultimele seturi de etichete de listă acceptat de HTML 4 sunt `<DIR>` și `<MENU>`, inițial proiectate pentru listarea directoarelor și meniurilor utilizatorului. Ele au fost însă utilizate tot mai rar, ajungând să fie nerecomandate în standardul HTML 4.

SUGESTIE

Deoarece `<DIR>` și `<MENU>` sunt nerecomandate, W3C încurajează folosirea etichetei `<UL>` în locul lor.

Atât `<DIR>` cât și `<MENU>` folosesc eticheta `<LI>` pentru a marca fiecare articol în parte, cu excepția listei de definiții. Sintaxele celor două elemente sunt identice cu cea pentru o listă neordonată (acest lucru este o parte din motivul pentru care sunt expuse unei eventuale excluderi din standardul HTML):

```
<MENU> <!-- or DIR -->
<LI>ListItem1
<LI>ListItem2
<LI>ListItemN
</MENU> <!-- or /DIR -->
```

Amândouă articolele sunt de obicei reproduse de browser ca liste neordonate. (Vezi Figura 8.11.) Rețineți că în Mosaic stilurile caracterelor sunt un pic diferite. Precedenta recomandare a W3C includea o listă de directoare pe mai multe coloane ca redare a etichetei `<DIR>`, deși browserele au urmat foarte rar această recomandare. Pentru `<MENU>` a fost recomandată o listă de opțiuni, cum ar fi acelea folosite pentru propunerea unui formular sau pentru alegerea unui răspuns la o întrebare. Această etichetă este de obicei redată identic cu o listă neordonată. Nici un articol nu are atribute, în afara des ignoratului și acum nerecomandatul `compact`.

OBSERVAȚIE

Pot fi folosite atributele standard pentru elementele `<DIR>` și `<MENU>`:

- `id`, `class`, `style`, `title` (Capitolul 6)
- `lang`, `dir` (Capitolul 40)
- `onclick`, `ondblclick`, `onmousedown`, `onmouseup`, `onmouseover`, `onmousemove`, `onmouseout`, `onkeypress`, `onkeydown`, `onkeyup` (Capitolul 28)

OBSERVAȚIE

Pot fi folosite elementele standard de formatare în cadrul acestor elemente:

- **elemente de font:** `TT`, `I`, `B`, `U`, `S`, `STRIKE`, `BIG`, `SMALL` (Capitolul 17)
- **elemente de frază:** `EM`, `STRONG`, `DFN`, `CODE`, `SAMP`, `KBD`, `VAR`, `CITE`, `ACRONYM` (Capitolul 7)

Pentru un cod mai clar și revizuirile ulterioare, asigurați-vă că orice etichetă de formatare deschisă într-un articol din listă este închisă, înainte ca următorul articol să înceapă.

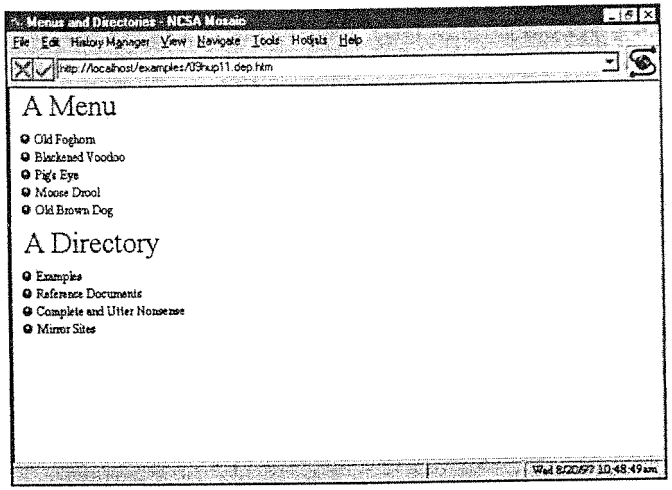


FIGURA 8.11. *Reprezentarea de bază a unui meniu sau a unei liste de directoare este tot o listă neordonată.*

Rezumat

Sunt două categorii principale de liste în HTML. Prima este lista ordonată, care poate fi folosită cu atributele sale pentru a crea o varietate de formate pentru informațiile dumneavoastră. A doua este lista neordonată, cu opțiunile sale pentru afișarea informațiilor fără priorități.

Implementarea reală a atributelor articolelor din listă variază în funcție de browser, așadar fiți atent la diferențe. În Netscape Navigator, de exemplu, fixarea unui atribut `<LI>` în mijlocul unei liste afectează și următoarele articole ale listei. Acest comportament nu există în Microsoft Internet Explorer.

Definițiile sunt alt tip de listă. Ele sunt unice prin faptul că nu necesită o etichetă de articol de listă, dar în schimb folosesc o pereche de etichete – un termen de definire și definiția în sine – pentru a potrivi articolele pereche din listă.

Două etichete adiționale, acum practic depășite, dar în continuare parte din standard, sunt listările de meniuri și de directoare. Implementarea lor favorită este ignorată de majoritatea browserelor, care au tendința de a le reprezenta pe cele două ca simple liste neordonate.

Următoarele două capitole, „Organizarea datelor cu ajutorul tabelelor" și „ Macheta și prezentarea tabelelor", adaugă mai multe opțiuni pentru afișarea textului. Cu etichetele de formatare a textului din capitolul anterior și tipurile de liste din acest capitol, vă dezvoltați un adevărat arsenal pentru a vă atinge scopurile prin intermediul HTML-ului 4.

Capitolul 9
Organizarea datelor cu ajutorul tabelelor

de Rick Darnell

În acest capitol:

- **Organizarea unui tabel simplu 116**
- **Gruparea rândurilor și coloanelor 124**
- **Afișarea datelor reale cu un tabel 129**

În lumea Webului, tabelele HTML au devenit unul dintre cele mai puternice și mai folosite instrumente pentru formatarea paginilor Web. În acest capitol, veți învăța despre scopul original al tabelelor, acela de a formata mai ușor datele tabulare, cum ar fi informațiile din foile de calcul tabelar sau cele din bazele de date.

Pe măsură ce începeți să explorați nuanțele etichetelor și ale atributelor acestora în acest capitol, probabil că veți descoperi și alte aplicații. În capitolul următor, „Macheta și prezentarea tabelelor", veți folosi tabele pentru a crea o varietate de machete de pagini care să nu fie acceptate direct de HTML.

Un dezavantaj este acela că etichetele de tabel sunt cele mai complicate și greu de folosit. În spațiul liniar al codului unei pagini HTML, trebuie să definiți un formular bidimensional. Este greu să vedeți care rând aparține cărei informații și cum o anume celulă va arăta în forma sa finală. Un browser activat pentru tabele este imperativ necesar, dacă nu folosiți un editor HTML vizual.

ATRIBUTE COMUNE

Majoritatea elementelor din acest capitol au atribute HTML comune, cum ar fi articolele care verifică identificarea, stilul, limbajul și tratarea evenimentelor. Deoarece atributele comune sunt atât de des întâlnite în HTML 4 (de aceea sunt numite „comune"), veți găsi o prezentare a lor în Introducere, la începutul acestei cărți. Puteți de asemenea găsi mai multe informații despre aceste caracteristici des întâlnite și elementele lor asociate în Anexa A, „HTML 4.0 – Referințe".

Între timp, în acest capitol, mă voi referi la aceste atribute comune folosind următoarele cuvinte cheie: *core* (principale), *international* (internaționale) și *events* (evenimente).

În plus, multe elemente pot accepta alte elemente în cadrul granițelor lor. Acestora li se va spune *elemente inline* și includ următoarele opțiuni:

Elemente `font`: Aceste elemente includ etichetele pentru bold (`<B>`) și italic (`<I>`) și sunt discutate în Capitolul 17, „Stil simplu cu text".

Elemente `phrase`: Aceste elemente includ etichete cum ar fi evidențiere (`<EM>`), cod (`<CODE>`) și adresă (`<ADDRESS>`) și sunt discutate în acest capitol.

Elemente `special`: Această colecție este o amestecătură de elemente care include următoarele: miniaplicații și obiecte (Capitolul 16, „Obiecte înglobate în HTML"), imagini (Capitolul 11, „Adăugarea imaginilor pe pagina dumneavoastră Web"), ancore (Capitolul 13, „Hiperlegături și ancore"), hărți imagine (Capitolul 14, „Crearea meniurilor cu hărți imagine"), scripturi (Capitolul 23, „Scurtă privire asupra limbajului JavaScript" și 24,"Folosirea limbajului VBScript"), fonturi, superscripturi și subscripturi (Capitolul 17), rânduri de pauză și citate scurte (în acest capitol) și cadre inline (Capitolul 20, „Crearea cadrelor").

(continuare)

(continuare)
Elementele `form`: Acest set de elemente este folosit pentru a adăuga elemente unui formular, inclusiv butoane, câmpuri de text și casete de validare.

Organizarea unui tabel simplu

Ca și alte elemente de bloc discutate în capitolele anterioare, un tabel este marcat la început și la sfârșit cu eticheta `<TABLE>` și conține o varietate de definiții ale rândurilor și coloanelor. La nivelul său cel mai simplu, un tabel constă din etichetele `<TABLE>`, un rând și o celulă.

```
<TABLE>
<TR> <!-- table row -->
<TD>Content <!-- table data -->
</TABLE>
```

Acest formular simplu nu are nici o utilizare practică. Totuși, ne permite să subliniem câteva lucruri. Mai întâi, eticheta `<TABLE>` trebuie închisă cu `</TABLE>`, altfel browserul nu va ști când să oprească formatarea tabelului. Apoi, rândul de tabel (`<TR>`) și celulele de date ale tabelului (`<TD>`) pot accepta etichetele corespunzătoare de închidere, dar acestea nu sunt necesare. Un nou rând sau o nouă celulă de date marchează automat sfârșitul elementului de dinainte (rând sau celulă).

SUGESTIE

Dacă aveți de gând să formatați conținutul rândurilor și celulelor dumneavoastră prin folosirea unei foi de stil, va fi nevoie să folosiți etichetele de închidere pentru elementele acelea. O foaie de stil nu lucrează cu o etichetă de deschidere-închidere.

După cum probabil că vă așteptați, sunt câteva atribute care se aplică acestor trei etichete de bază ale tabelelor. Vom începe cu `<TABLE>`. În plus față de atributele *principale, internaționale* și *evenimente*, un tabel are următoarele opțiuni:

- **`Align`:** Acesta este atributul standard de aliniere, care acceptă valorile `left`, `right` sau `center` pentru a controla plasamentul pe o pagină. Majoritatea browserelor au prestabilită alinierea la stânga sau centrală.
- **`Width`:** Lățimea tabelului, exprimată în pixeli sau ca procentaj din lățimea ferestrei browserului. (Vezi Figura 9.1.) De exemplu, un tabel care ocupă jumătate din lățimea ferestrei disponibile este marcat cu `width=50%`, iar un tabel lat de 500 de pixeli este specificat cu `width=500`. Valoarea prestabilită este doar atât de mare încât să permită includerea celui mai lung conținut în fiecare coloană.

  Ca regulă generală, folosiți valoarea procentuală pentru a asigura compatibilitatea pe o gamă mai mare de afișaje și platforme. Valoarea în pixeli este folositoare când aveți nevoie de o lățime fixă pentru conținuturi cum ar fi imaginile.

- **`Cols`:** Acest atribut este nou în HTML 4. El comunică browserului câte coloane sunt incluse în tabel. În trecut, browserul trebuia să aștepte până când era încărcat întregul tabel pentru a începe redarea sa.
- **`Frame`:** Acest atribut, înrudit cu atributul `border`, determină unde ar trebui să apară cadrul care înconjoară tabelul. Valorile sale posibile sunt `void` (nici unul), `above` (partea superioară), `below` (partea inferioară), `hsides` (părțile orizontale), `lhs` (partea stângă), `rhs` (partea dreaptă), `vsides` (părțile verticale) și `box` sau `border` (toate părțile).
- **`Border`:** Acest atribut setează lățimea unui chenar în jurul tabelului, în pixeli. Valoarea prestabilită variază în funcție de browser, dar de obicei este de 2 sau 3 pixeli. Dacă nu doriți un chenar, folosiți valoarea `0`.

- **`Rules`:** Acest atribut determină unde sunt folosite linii pentru separarea celulelor în cadrul tabelului. Valorile sale posibile sunt `none` (nici o linie), `groups` (între grupurile logice cum ar fi `TBODY` și `THEAD`), `rows` (între rânduri), `cols` (între coloane) și `all` (între toate celulele).
- **`Cellspacing`:** Identificat în pixeli, acest atribut determină mărimea spațiului dintre celule. Ca efect secundar, câteodată mai determină și lățimea liniei dintre celule.
- **`Cellpadding`:** Asemănător cu `cellspacing` și de asemenea identificat în pixeli, acest atribut stabilește mărimea spațiului dintre marginea celulei și conținutul său. Diferența dintre cele două este arătată în Figura 9.2.

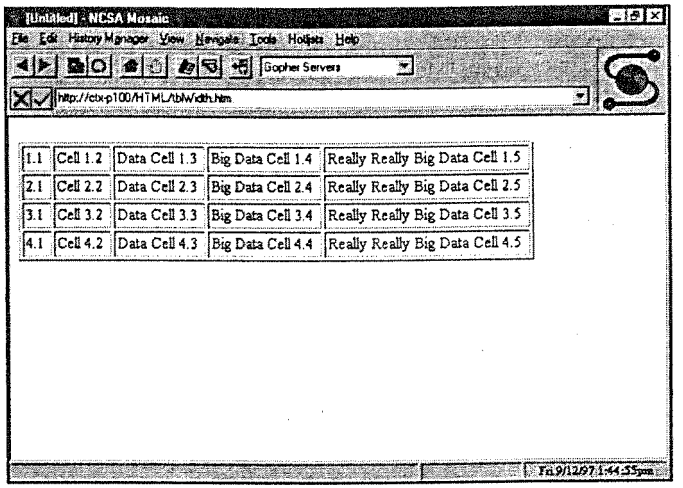


FIGURA 9.1. *Fiecare coloană a tabelului este potrivită ca mărime pentru a încăpea cel mai lat element.*

După definirea parametrilor de bază ai unui tabel, în următoarele două secțiuni vor fi prezentate etichetele pentru rânduri și date.

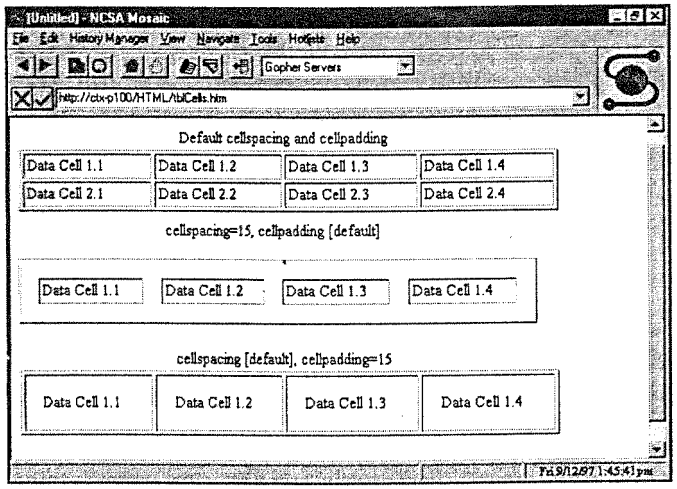


FIGURA 9.2. *Aceste variante de tabel includ o versiune cu valorile prestabilite, una cu spațierea celulelor mărită și alta cu spațiul liber din celulă mărit.*

Unde începe rândul: <TR>

Începutul rândului este marcat prin folosirea etichetei <TR> cu sintaxa următoare:

```
<TR [align=horizontalAlign][valign=verticalAlign]
    [core][international][events]>
```

<TR> marchează începutul rândului de definiții de celule și poate fixa setările prestabilite de afișare ale celulelor sale. Atributul `align` fixează spațierea orizontală a celulelor de pe rând și poate avea una dintre următoarele valori: `left`, `right`, `justify` sau `char`.

OBSERVAȚIE

Ultima valoare, `char`, este o setare specială care formatează conținutul în jurul unui anume caracter. Valoarea prestabilită pentru browserele în limba engleză este punctul zecimal pentru formatarea informațiilor numerice, dar browserele în limba franceză folosesc virgula. Puteți folosi atributul `char` pentru a fixa caracterul la o altă valoare, cum ar fi `char="*"`.

Atributul `valign` fixează plasarea verticală a informațiilor în celulă și poate avea una dintre valorile: `top`, `middle` sau `bottom`. Eticheta <TR> nu necesită o etichetă de închidere, dar puteți folosi una pentru a face granițele fiecărui rând mai ușor de identificat. În plus, atributul <TR> poate accepta atributele core, international și event.

Există două atribute pentru <TR> care fixează comportamentul tuturor celulelor din rând, dar celulele individuale pot ignora valorile lui <TR>. Primul atribut este `align`, care este asemănător cu `align`-ul din eticheta <TABLE>, dar este limitat pentru a controla alinierea orizontală a materialului în celule. Valorile disponibile sunt `left` (valoarea prestabilită), `right` sau `center`. Figura 9.3 arată un tabel care conține aceste trei opțiuni.

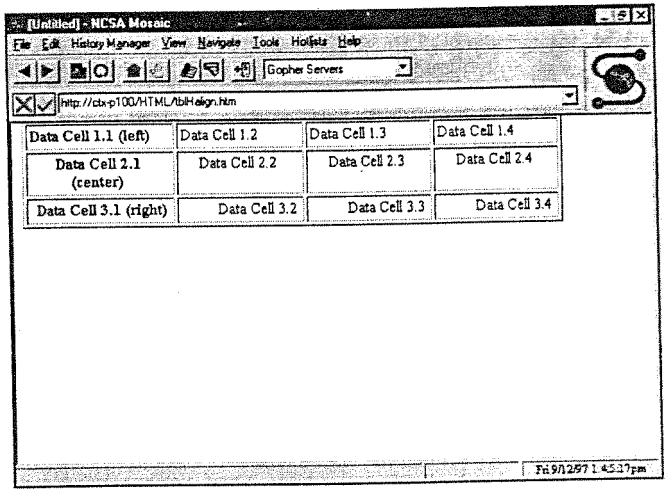


FIGURA 9.3. *În acest tabel, fiecare rând este fixat pentru o aliniere orizontală diferită, prin folosirea atributelor de aliniere <TR>.*

Definirea celulelor tabelului: <TD> și <TH>

După ce am marcat începutul rândului cu <TR>, a venit timpul să ne apucăm, în cele din urmă, de umplerea fiecărei celule. Sunt două tipuri de etichete de celule folosite în acest tabel. Prima, <TH>, marchează o celulă antet și este similară unei etichete de titlu pe o pagină Web. Cealaltă este o etichetă de celulă de date, <TD>, folosită pentru corpul unui tabel.

După ce este identificat începutul rândului, puteți începe adăugarea conținutului fiecărei celule. Aceasta este sintaxa pentru <TD>:

```
<TD [rowspan=numRows][colspan=numCols][axis=headerName] [axes=axisNames]
[alignment][nowrap][core] [international][events]>cellContent
```

Atributele pentru <TD> sunt aprofundate în secțiunile următoare. Etichetele de închidere </TD> sau </TH> nu sunt cerute pentru celulele individuale, dar mulți designeri HTML și programe de editare le folosesc pentru claritate.

OBSERVAȚIE

În funcție de numărul de celule și conținutul lor, există două convenții pentru plasamentul etichetei <TD>. Primul plasament este pe același rând cu eticheta container <TR>, ceea ce înseamnă că veți avea o linie HTML începând cu <TR> pentru fiecare rând din documentul dumneavoastră.

Totuși, dacă aveți multe celule sau dacă acestea includ articole cum ar fi hiperlegături sau imagini, probabil veți dori să plasați fiecare definiție de celulă pe propriul său rând, cu o etichetă de închidere </TR> și un carriage return suplimentar la sfârșitul fiecărui rând.

Indiferent de metoda de plasare, codul se comportă la fel. Depinde de dumneavoastră să alegeți formatul care să ajute la clarificarea codului în vederea editărilor și revizuirilor ulterioare.

Când vă gândiți la opțiunile de formatare pentru fiecare celulă în parte, gândiți-vă la etichetele <TD> și <TH> ca fiind etichete <BODY>. Orice etichetă validă pentru corpul unui document HTML este validă și într-o celulă, ceea ce înseamnă că puteți îngloba text, grafică, formulare, plug-ins-uri și obiecte, miniaplicații și chiar alte tabele. Această caracteristică este ceea ce dă tabelelor puterea când sunt folosite la formatarea unei pagini.

Atributele pentru celulele tabelului sunt împărțite în două mari categorii: acelea care afectează mărimea celulelor și acelea care afectează conținutul celulelor.

Un tip special de celule de date din tabele este celula antet, <TH>. Majoritatea browserelor includ un stil de font diferit pentru celulele antet, astfel încât să ajute la evidențierea scopului lor pentru utilizator. (Vezi Figura 9.4.)

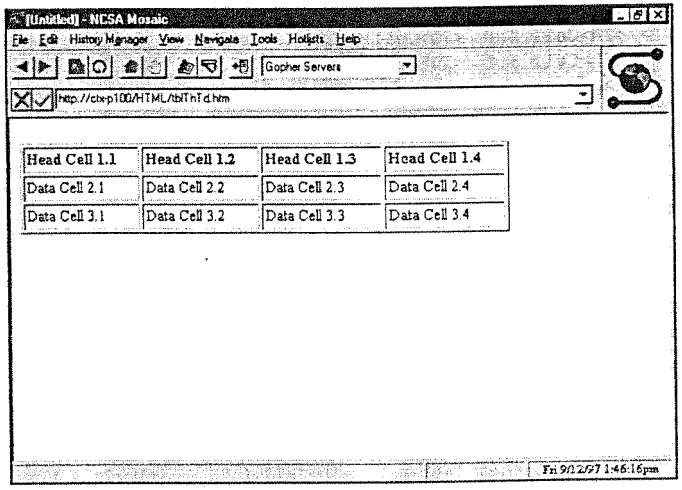


FIGURA 9.4. *Primul rând este definit cu eticheta <TH>, iar următoarele, cu eticheta <TD>.*

SUGESTIE

Celulele antet sunt o definiție logică, nu una fizică. Interpretarea etichetei <TH> este lăsată în voia browserului și variază în funcție de platforme și aplicații.

Sintaxa pentru <TH> este:

```
<TABLE>
<TR><TH TDattributes>Cell Header
</TABLE>
```

În această etichetă, *Call Header* este conținutul celulei antet, iar *TDattributes* reprezintă oricare sau chiar toate atributele valabile pentru <TD>.

Atributele de mărime ale celulei: rowspan și colspan

Primul set de atribute determină capacitatea celulei de a fuziona cu cele învecinate. Atributele rowspan și colspan sunt folosite pentru a combina celule adiacente în celule mai mari. Este important să observați că atunci când folosiți aceste atribute, celulele adiacente nu sunt eliminate; ele sunt doar „ascunse" câtă vreme celulele acaparatoare le folosesc spațiul. Sintaxa este următoarea:

```
<TD rowspan=numRows colspan=numCols >
```

În acest atribut, *numRows* este numărul de rânduri, inclusiv celula curentă, care sunt unite. În același fel, *numCols* este numărul de coloane alăturate. Valoarea prestabilită pentru amândouă este 1.

Alipirea este un pic mai greu de planificat și de realizat. De exemplu, începeți cu un tabel 3X3, care are trei rânduri cu câte trei celule fiecare. (Vezi Listingul 9.1.)

Listingul 9.1. Un cod HTML simplu pentru afișarea unui tabel 3X3.

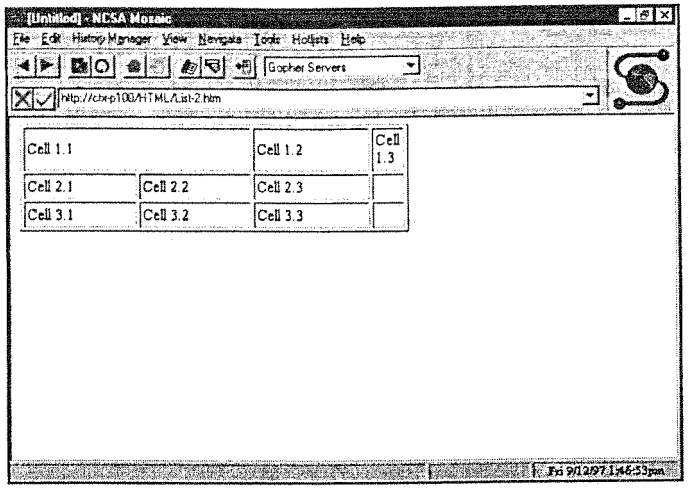
```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0//EN">
<HTML>
<BODY>
<TABLE border=3 width=75%>
 <TR> <TD>Cell 1.1  <TD>Cell 1.2  <TD>Cell 1.3
 <TR> <TD>Cell 2.1  <TD>Cell 2.2  <TD>Cell 2.3
 <TR> <TD>Cell 3.1  <TD>Cell 3.2  <TD>Cell 3.3
</TABLE>
</BODY>
</HTML>
```

Acum puteți începe fuzionarea celulelor prin combinarea celulelor 1.1 și 1.2. Deoarece această metodă se aplică pe coloane, este numită *alipirea coloanelor*. (Vezi Listingul 9.2 și Figura 9.5.) Pentru a face acest lucru, este adăugat colspan=2 ca atribut al etichetei de date de tabel pentru celula 1.1.

Listingul 9.2. Alipirea coloanelor peste celulele 1.1 și 1.2 prin adăugarea atributului colspan.

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0//EN">
<HTML>
<BODY>
<TABLE border=3 width=75%>
 <TR> <TD colspan=2>Cell 1.1  <TD>Cell 1.2      <TD>Cell 1.3
 <TR> <TD>Cell 2.1            <TD>Cell 2.2      <TD>Cell 2.3
 <TR> <TD>Cell 3.1            <TD>Cell 3.2      <TD>Cell 3.3
</TABLE>
</BODY>
</HTML>
```


FIGURA 9.5.
Tabelul rezultat în urma alipirii coloanelor.



Observați cele două rânduri de celule de jos în Figura 9.5. După cum puteți vedea, alipirea coloanelor nu înseamnă doar adăugarea atributelor span. Iată ce s-a întâmplat. Când browserul a întâlnit colspan=2, a știut că celula 1.1 avea nevoie să ocupe spațiul pentru două celule, și a furnizat spațiul corespunzător. Apoi, a continuat de-a lungul rândului și a terminat adăugarea celor două celule. Rezultatul a fost un spațiu pentru un total de patru celule (două unite și două individuale).

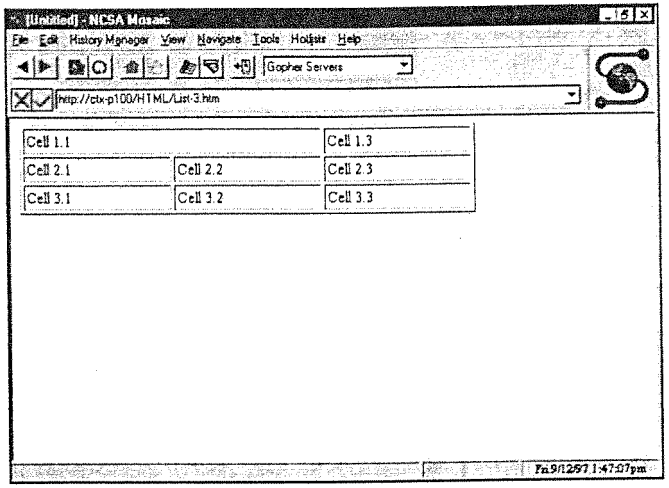
La noua etichetă de rând, browserul a început un rând nou prin adăugarea a trei celule. Browserul nu a întâlnit a patra etichetă <TD>, deși era loc în tabel pentru aceasta. În loc să adauge o celulă pe care utilizatorul nu a specificat-o, el a umplut locul cu spații goale. Rezultatul final este un tabel 3X4 cu cele două rânduri de jos doar parțial definite.

Această caracteristică a alipirii celulelor poate cauza multe dureri de cap când lucrați cu tabele. Pentru fiecare celulă alipită, trebuie să înlăturați definiția corespunzătoare celulei respective. (Vezi Listingul 9.3 și Figura 9.6.)

Listingul 9.3. O alipire a coloanelor peste celulele 1.1 și 1.2 necesită adăugarea atributului colspan și înlăturarea etichetei celulei a doua.

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0//EN">
<HTML>
<BODY>
<TABLE border=3 width=75%>
 <TR> <TD colspan=2>Cell 1.1                    <TD>Cell 1.3
 <TR> <TD>Cell 2.1            <TD>Cell 2.2      <TD>Cell 2.3
 <TR> <TD>Cell 3.1            <TD>Cell 3.2      <TD>Cell 3.3
</TABLE>
</BODY>
</HTML>
```

FIGURA 9.6
Cu a doua celulă de date înlăturată, tabelul păstrează configurația sa originală, de 3X3.



SUGESTIE

Ar putea fi un pic cam greoi să editām tabelele cu metodele grafice prezentate în Listingul 9.2 și 9.3, dar este folositor să determinăm care celule ar trebui înlăturate pentru alipire.

Aceeași regulă se aplică pentru atributul `rowspan`, exceptând faptul că trebuie înlăturate celulele de sub eticheta originală. În Listingul 9.4, de exemplu, sunt înlăturate trei etichete alăturate. De asemenea, este posibilă combinarea a două atribute pentru a obține alte efecte. (Vezi Figura 9.7.)

Listingul 9.4. Se combină prima celulă cu rândul și coloana următoare.

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0//EN">
<HTML>
<BODY>
<TABLE border=3 width=75%>
 <TR> <TD colspan=2 rowspan=2>Cell 1.1                    <TD>Cell 1.3
 <TR>                                                     <TD>Cell 2.3
 <TR> <TD>Cell 3.1                    <TD>Cell 3.2        <TD>Cell 3.3
</TABLE>
</BODY>
</HTML>
```

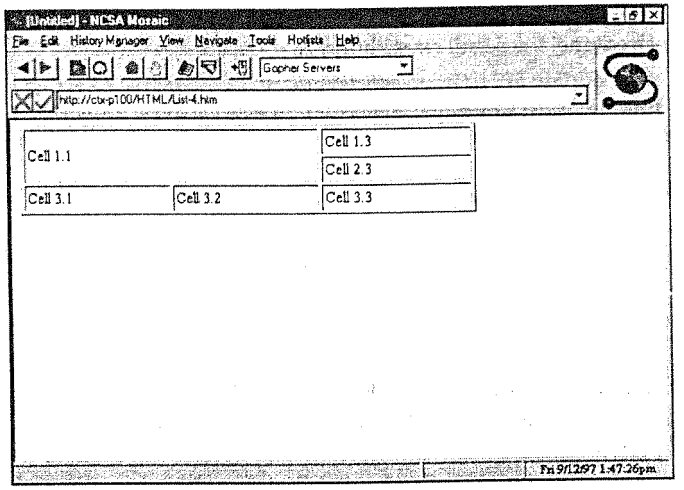
Când alipirea se face în două direcții deodată, trebuie să ștergeți toate celulele de date din calea fuzionării, ceea ce înseamnă înlăturarea celulelor din dreapta, stânga și de pe diagonală.

Identificarea celulelor antet: `axis` și `axes`

Atributele `axis` și `axes` comunică browserului care anteturi de rând sau de coloană se aplică unei anume celule. Iată cum sunt folosite:

```
<TD axis="cellName" axes="cellName1, cellName2"
```


FIGURA 9.7.
Pentru celulele de date învecinate au fost combinate într-o singură celulă mare, dar tabelul își păstrează aspectul original, de 3X3.



În aceste atribute, numele se referă la o altă celulă care se comportă ca un antet pentru ce actuală. În mod normal, numele este chiar valoarea atributului `id` pentru o celulă. Dac atributul `id` nu este folosit, atunci numele celulei va fi preluat din conținutul său.

Ca exemplu, priviți acest tabel:

```
<TABLE>
  <TR>
    <TH>Name <TH>Department <TH>Certification
  <TR>
    <TD axis="Beree" axes="Name">Russ Beree <TD>MFD <TD>Technician
  <TR>
    <TD axis="Colwell" axes="Name">Bill Colwell <TD>MRFD <TD>Technician
  <TR>
    <TD axis="Lanier" axes="Name">Marty Lanier <TD>MRFD <TD>Technician
</TABLE>
```

Valoarea lui `axis` pentru prima celulă a tabelului este numele acelei celule, dar valoarea l `axes` se referă la coloană și este aceeași pentru fiecare celulă din acea coloană.

Deși nu sunt încă acceptate de cele mai cunoscute browsere, aceste două atribute au fo incluse în suportul general de bază de date pentru browsere.

Alinierea conținutului celulelor: `align` și `valign`

Următorul grup de atribute controlează modul în care textul sau alt conținut este plasat celulă. Sintaxa pentru aceste atribute este:

```
<TD align=left|center|right valign=top|middle|bottom nowrap>
```

În aceste atribute, `align` controlează alinierea orizontală, `valign` pe cea verticală, iar nowr forțează textul conținut într-o celulă să rămână pe un singur rând.

Valorile și comportamentul pentru `align` și `valign` sunt aceleași cu ale etichetei `<TR>` anulează valorile acesteia doar pentru celula curentă.

Alt atribut, `nowrap`, anulează aranjarea automată a cuvintelor în cadrul granițelor celulel în funcție de alte setări ale tabelului, cum ar fi lățimea fixă sau proporțională a unei celu

sau a întregului tabel. Este un atribut nerecomandat în HTML 4.0, acum fiind preferată folosirea unei foi de stil cu atributul `white-space`.

Dacă tabelul nu are o lățime specificată, celula afectată se va extinde pe orizontală pentru a permite textului să încapă. Același lucru este valabil și atunci când lățimea tabelului este specificată ca procentaj din fereastra browserului; în acest caz, lățimea tabelului se extinde pentru a putea conține celula mai mare.

Dacă este folosit atributul `nowrap` împreună cu orice setări `width` ale lățimii în pixeli, setările `width` au prioritate, iar înfășurarea liniei este reluată.

Un efect asemănător este produs prin excluderea lui `nowrap` și folosirea spațiilor solide (` `) între cuvintele care ar trebui să rămână la un loc pe același rând. Această tehnică vă permite să păstrați anumite cuvinte legate și le lasă pe celelalte să curgă pe un rând nou, după cum este nevoie.

Atribuirea unui nume tabelului: <CAPTION>

Ultimul dintre elementele de bază pentru un tabel este un titlu descriptiv pentru a-l identifica. Eticheta `<CAPTION>` este pentru un tabel ceea ce este eticheta `<TITLE>` pentru `<HEAD>`-ul documentului, cu excepția că puteți specifica dacă denumirea va apărea deasupra sau dedesubtul tabelului.

Dacă este inclusă, denumirea trebuie să apară imediat după eticheta `<TABLE>` de deschidere, înainte de orice definiție de rând sau celulă a tabelului. Sintaxa este:

```
<TABLE>
<CAPTION [align=alignment][core][international][attributes]>
captionText
</CAPTION>
...table content...
</TABLE>
```

Atributul `align` este folosit un pic diferit față de alte elemente. În acest caz, el indică unde este plasată denumirea față de tabel. Valorile sale posibile sunt `top`, `bottom`, `left` sau `right`.

Gruparea rândurilor și coloanelor

HTML 4.0 include acum suport pentru grupuri de rânduri și de coloane pentru a defini mai bine structura tabelului. Încă neacceptate pe larg, aceste etichete noi pentru tabel vor ajuta browserul să afișeze material de identificare esențial în cadrul celulelor și vor da autorilor mai multă flexibilitate în structurarea informațiilor.

Gupuri de rânduri: <THEAD>, <TBODY> și <TFOOT>

Etichetele pentru gruparea rândurilor – `<THEAD>`, `<TBODY>` și `<TFOOT>` – probabil că vor fi folosite des de acei oameni care au nevoie să afișeze date tabulare și alte feluri de informații structurate pe rânduri. Iată sintaxa pentru folosirea celor trei etichete:

```
<TABLE>
<THEAD [alignment][core][international][events]>
...rows and cells...
<TFOOT [alignment][core][international][events]>
...rows and cells...
<TBODY [alignment][core][international][events]>
...rows and cells...
</TABLE>
```


Folosirea etichetelor de deschidere și de închidere pentru grupuri de rânduri se bazează mult pe inferența celorlalte părți ale tabelului. De exemplu, `</THEAD>` nu este cerut dacă este urmat de un `</TFOOT>` sau un `</TBODY>`. Același lucru este valabil și pentru `</TFOOT>`. Dacă este folosit `<THEAD>` sau `<TFOOT>`, atunci trebuie să specificați începutul corpului de conținut cu `<TBODY>`. Dacă nu este folosit nici `<THEAD>` și nici `<TFOOT>`, atunci `<TBODY>` este presupus și nu aveți nevoie de o etichetă de deschidere sau de închidere.

Modul în care un browser folosește un grup va depinde de cei ce au creat browserul, dar una dintre cele mai des întâlnite variante este aceea în care rândurile `<THEAD>` și `<TFOOT>` rămân vizibile pe ecran în timp ce utilizatorul desfășoară `<TBODY>`-ul în sus și în jos. Mai există și posibilitatea ca browserul să atribuie aspectul prestabilit antetului și subsolului tabelului, deși standardul W3C nu face nici o recomandare asupra unui mod preferat de afișare.

SUGESTIE

Datorită posibilei întrebuințări în afișarea unui tabel, `<TFOOT>` trebuie să apară înainte de `<TBODY>`. Această poziționare oferă browserului ocazia să încarce și să evalueze ce să facă în privința subsolului înaintea procesării restului de tabel.

Grupuri de coloane: <COLGROUP> și <COL>

`<COLGROUP>` și `<COL>` reprezintă altă încercare de a ajuta la definirea mai ușoară a unor părți ale tabelului. Înainte de aceste etichete, trebuia să țineți minte numărul de celule `<TD>` din fiecare rând pentru a ști câte coloane sunt în tabel, care celule sunt în aceeași coloană și cum să formatați celulele într-o coloană dată.

Sintaxa pentru eticheta de grup de coloane este următoarea:

```
<COL [span=][width=][alignment][core][international][events]>
```

Pentru `<COLGROUP>`, sintaxa este:

```
<COLGROUP [span=][width=][alignment][core][international][events]>
```

Eticheta de închidere este opțională pentru `<COLGROUP>`, deși aceasta este recomandată până în momentul în care browserele vor începe să o recunoască. Altfel, browsere cum ar fi Navigator 4.02 pot alege să nu afișeze deloc restul tabelului. Fratele său, `<COL>`, este un element gol, așa că nu folosește o etichetă de închidere.

Puteți include o serie de coloane într-un grup de coloane pentru identificarea ușoară și anularea atributelor. De exemplu, acest tabel include definiții posibile pentru cinci coloane:

```
<TABLE>
  <COL>
  <COL>
  <COL>
  <COL>
  <COL>
  <TR>
    ...table contents...
</TABLE>
```

Dacă doriți să le puneți pe primele trei într-un grup și să le formatați în același fel, veți folosi `<COLGROUP>` în modul următor:

```
<TABLE>
  <COLGROUP width="10%">
    <COL>
    <COL>
```

```
    <COL>
  </COLGROUP>
  <COL>
  <COL>
  <TR>
    ...table contents...
</TABLE>
```

Prin definiție, fiecare tabel are cel puțin o coloană, chiar dacă nu este nevoie de o etichetă <COL> pentru a o identifica. Atributul span al etichetei coloanei identifică câte coloane sunt în grupul actual, având prestabilită valoarea 1. Atributele width și align fixează înfățișarea de bază a celulei coloanei. În exemplul precedent, puteați omite cele trei etichete de coloane din grupul de coloane și să adăugați atributul span:

```
<TABLE>
  <COLGROUP span=3 width="10%">
  </COLGROUP>
  <COL width="40%">
  <COL>
  <TR>
    ...table contents...
</TABLE>
```

SUGESTIE

Dacă nu sunt măcar cinci etichete <TD> sau echivalentul lor (<TD colspan=5>) în corpul documentului, atunci etichetele de coloane suplimentare sau alipirea grupului de coloane vor fi ignorate.

Atributul width nu mai este asociat cu celulele individuale. Prin translarea sa către un element de coloană, aveți acum trei modalități de a specifica lățimea celulei: absolută, ca procentaj și relativă.

În prima metodă, *absolută*, este specificată de un singur număr întreg, care indică numărul de pixeli. Această metodă este folositoare la adăugarea articolelor cum ar fi imaginile și alte elemente care necesită mărimi exacte, dar nu este recomandată uzului general deoarece este imposibil de știut mărimea ariei de afișare a utilizatorului. A doua metodă, *procentaj*, definește lățimea coloanei ca procent din lățimea totală a tabelului și este identificată ca un număr întreg de la 0 la 100 urmat de un semn de procent (%).

Ultima opțiune, *relativă*, este o metodă nouă de a defini lățimea, prin atribuirea unei proporții din lățimea disponibilă. Priviți următoarea specificare a unui grup de coloane:

```
<COLGROUP width="1*">
<COLGROUP width="4*">
<COLGROUP width="1*">
<COLGROUP width="2*">
```

Acest set de patru etichete definește patru coloane cu o lățime proporțională de 8 unități. Prima și a treia coloană sunt 1/8 din lățimea tabelului, a doua este 4/8 și ultima este 2/8. Dacă lățimea totală a tabelului este de 100 de procente din fereastra browserului, atunci prima și a treia coloană ar fi câte 12,5 procente, a doua ar fi 50 de procente și a treia 25 de procente. O valoare specială pentru această metodă este 0*, care arată că browserul ar trebui să folosească cel mai mic spațiu posibil ca să găzduiască conținutul cel mai lat dintr-o coloană.

Pentru a vedea de ce este bine să adăugați <COLGROUP> la cutia dumneavoastră de unelte, priviți acest tabel construit după specificațiile HTML 3.2:


```
<TABLE>
  <TR>
    <TD width="20%">content <TD width="20%">content <TD width="20%">content
    <TD width="10%">content <TD width="10%">content
    <TD width="10%">content <TD width="10%">content
  <TR>
    <TD width="20%">content <TD width="20%">content <TD width="20%">content
    <TD width="10%">content <TD width="10%">content
    <TD width="10%">content <TD width="10%">content
</TABLE>
```

Pentru a vă asigura că fiecare coloană își menține proporțiile, trebuie să includeți atributul width pentru fiecare celulă de date. Folosind <COLGROUP>, dacă adăugați două rânduri la începutul tabelului care să identifice setările coloanei, puteți să eliminați atributul width din corpul tabelului și să faceți rezultatul mult mai ușor de citit:

```
<TABLE cols="7">
  <COLGROUP span="3" width="20%">
  <COLGROUP span="4" width="10%">
<TR>
  <TD>content <TD>content
  <TD>content <TD>content <TD>content
<TR>
  <TD>content <TD>content
  <TD>content <TD>content <TD>content
</TABLE>
```

După ce am învățat toate uneltele necesare pentru construirea tabelelor, este timpul să aruncăm o privire de ansamblu, să vedem cum se împacă toate etichetele și atributele.

A cui setare prestabilită o anulează pe a cui?

După cum ați văzut, în etichetele de dinaintea etichetelor propriu-zise de celule de date au fost câteva ocazii să influențăm lățimea celulei. Deci, al cui atribut are întâietate?

Ca regulă generală, ultimul atribut întâlnit fixează standardul pentru restul tabelului, deși aceasta nu se întâmplă întotdeauna. Priviți următoarea definire a unui tabel:

```
<TABLE width=400>
<CAPTION>Table width=400, Cell width=500</CAPTION>
<TR><TD width=200>200 pixel-wide cell <TD width=300>300 pixel-wide cell
</TABLE>
```

În această definiție, lățimea tabelului este fixată la 400 de pixeli. Totuși, în definițiile primelor două celule de date, lățimea este forțată la 500 de pixeli. Ce se întâmplă? Ultima etichetă are prioritate în Mosaic, iar eticheta <TABLE> câștigă în Netscape.

Iată ce se întâmplă în Figura 9.8. Setarea width din eticheta <TABLE> fixează lățimea fizică. Când browserul întâlnește celulele de date cu atributul lor width, calculează totalul pixelilor necesari rândului. Când observă inegalitatea, decide că celulele individuale știu ce vor, deci ajustează lățimea totală a tabelului pentru a se potrivi cu acestea.

În Figura 9.9, Netscape face exact pe dos. Presupune că eticheta <TABLE> deține limitele absolute ale graniței exterioare, așa că micșorează primul tabel pentru a încăpea în spațiul atribuit, ignorând cererea din celulele de date.

Morala acestei povești? Asigurați-vă că toate corespund și nu veți avea nici o problemă. Dacă testați un tabel pe diferite browsere și vedeți rezultate cu diferențe majore între ele, atunci este cazul să vă întoarceți și să numărați din nou lățimile.

FIGURA 9.8.
Primul tabel ajunge să fie mai lat prin extinderea la suma lățimilor celulelor de date, dar al doilea nu are setări de lățime pentru celulele de date.

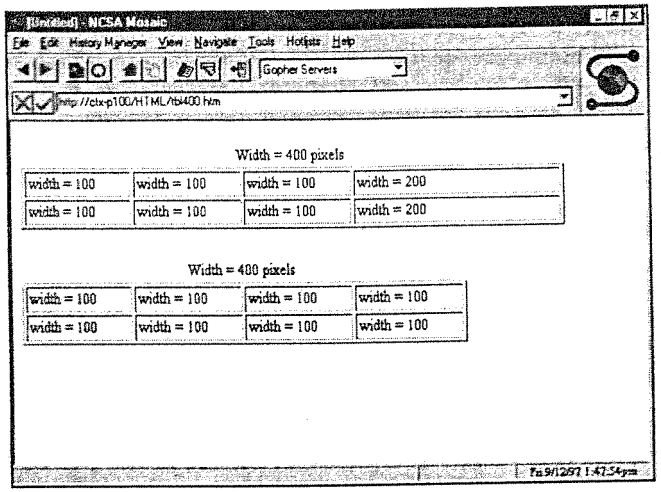
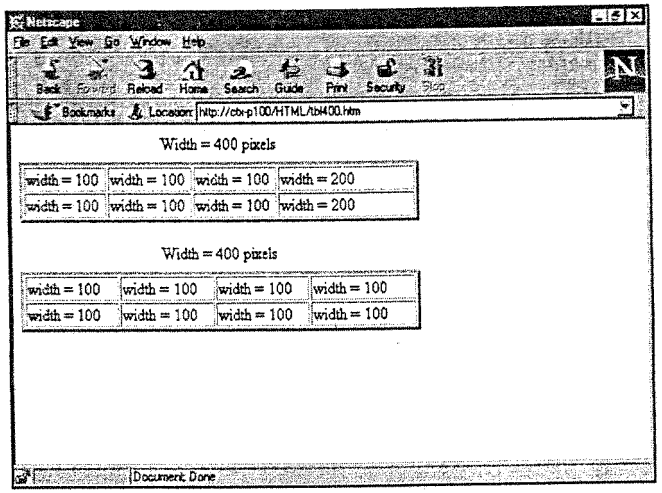


FIGURA 9.9.
În Netscape, primul tabel rămâne la 400 de pixeli, la fel ca al doilea.



Acum priviți alt exemplu – de data aceasta privind alinierea textului:

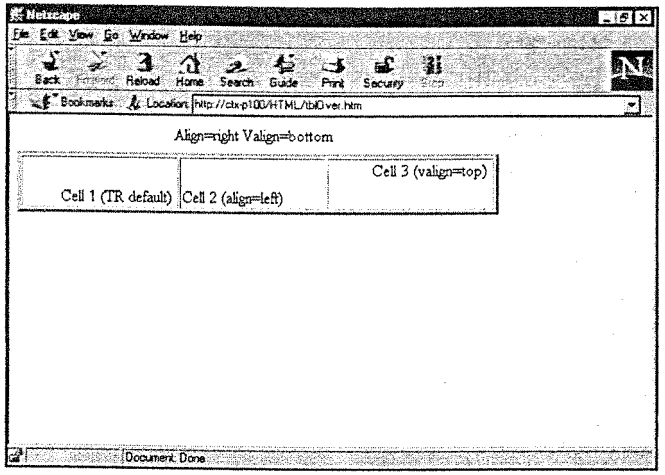
```
<TABLE>
<CAPTION>Align=right Valign=bottom</CAPTION>
<TR align=right valign=bottom>
  <TD>Cell 1 (TR default)
  <TD align=left>Cell 2 (align=left)
  <TD valign=top>Cell 3 (valign=top)
</TABLE>
```

Prima celulă din tabel acceptă valoarea prestabilită de la eticheta <TR>. Celula 2 anulează alinierea orizontală și forțează textul către stânga, iar celula 3 anulează alinierea verticală și forțează textul către vârf. (Vezi Figura 9.10.)

Observați că o celulă din rând nu moștenește nici o presetare de la celula anterioară. Fiecare celulă de date respectă setările definite în eticheta <TR>, până când un atribut este în mod special anulat în eticheta <TD>.

Din păcate, există multe excepții de la reguli. În acest caz, atributul `height` folosit în prima celulă de date setează de asemenea înălțimea pentru toate celelalte celule din rând. Dacă celulele au aceeași înălțime, se asigură integritatea chenarelor folosite la delimitarea lor. Dacă fiecare celulă ar avea propria înălțime, rezultatul ar fi un amestec de zigzaguri și trepte generate de liniile desenate în jurul fiecărei celule.

FIGURA 9.10.
Deși alinierea celulelor este fixată în dreapta-jos, fiecare dintre următoarele două celule anulează una din pozițiile prestabilite.



Afișarea datelor reale cu un tabel

Acum, că ați învățat sintaxa pentru tabele, este timpul să privim un exemplu practic despre cum este folosit un tabel de date pe World Wide Web.

Exemplul este un tabel simplu pentru afișarea datelor tabulare. (Vezi Figura 9.11.) În acest caz, este o listă de evenimente pentru un singur utilizator generată dintr-o bază de date cu un script CGI. Deși aceasta nu este o pagină generată de un om, formatul tabelului urmează standardul HTML. (Vezi Listingul 9.5.)

Listingul 9.5. Această pagină HTML a fost generată de un script CGI și include un tabel pentru afișarea informațiilor tabulare.

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0//EN">
<HTML>
<HEAD>
<META HTTP-EQUIV="Content-Type" CONTENT="text/html">
<TITLE>Incident Log</TITLE>
<BASE HREF=http://localhost/IncidentLog/>
</HEAD>
<BODY>
<H1>Volunteer Time Record</H1>
<TABLE BORDER="0" WIDTH="75%" CELLSPACING="0" CELLPADDING="0">
<CAPTION ALIGN="TOP">Prepared for the month of February 1997</CAPTION>
<TR ALIGN="CENTER" VALIGN="CENTER">
```

(continuare)

```
    <TD>
      <!-- BODY CELL -->
    </TD>
  </TR>
</TABLE>
```

Atributul `width` a fost adăugat etichetei `<TABLE>` pentru a se ocupa maximum de spațiu pe pagină. La urma urmelor, veți folosi acest tabel ca să formatați pagina, așa încât ar trebui să-și rezerve tot spațiul pe care îl poate ocupa. De asemenea, cei 15 pixeli din `cellspacing` au fost adăugați ca să se asigure un pic de spațiu între coloane, iar chenarul a fost ascuns pentru ca formatarea paginii să rămână invizibilă.

Acum trebuie să ajustați rândurile tabelului pentru ca acestea să-și afișeze corect conținutul. Veți dori ca lățimea coloanei de index să fie jumătate din lățimea coloanei de corp, iar conținutul să înceapă de la partea superioară a celulelor:

```
<TR valign="top">
  <TD width="33%">
    <!-- Empty cell to keep header from index column -->
  </TD>
  <TD width="67%" id="tblTitle">
    <!-- HEADER CELL -->
  </TD>
</TR>
<TR valign="top">
  <TD id="tblIndex">
    <!-- INDEX CELL -->
  </TD>
  <TD id="tblBody">
    <!--BODY CELL -->
  </TD>
 </TR>
```

A fost adăugat un atribut `id` pentru fiecare dintre celulele principale și celula antet, pentru cazul în care vă hotărâți să aplicați altă formatare sau foi de stil într-o revizuire ulterioară a documentului.

Adăugarea conținutului unei machete pe două coloane

Cu structura tabelului deja executată, putem începe să adăugăm conținutul. În stânga a fost plasată o listă de definiții care furnizează indexul conținutului, iar corpul de conținut a fost așezat în coloana din dreapta. (Vezi Listingul 10.1.)

Listingul 10.1. Carcasa tabelului este umplută cu conținut.

```
<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 4.0//EN">
<TABLE width=100% cellspacing=15 border=0>
  <TR valign="top">
    <TD width="33%">
    </TD>
    <TD width="67%" id="tblTitle">
    <H1>Backcountry Travel</H1>
    </TD>
  <TR valign="top">
    <TD>
    <DL>
    <DT><STRONG><A href="#intro" alt="Introduction"> Introduction </A></STRONG>
```



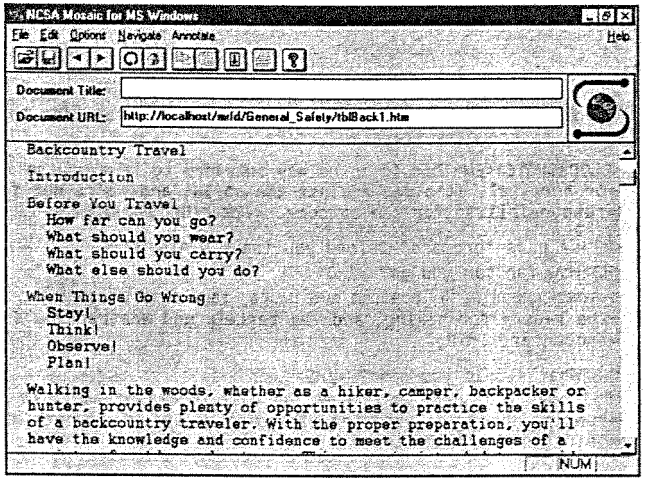
```
    </DL>
    <DL>
    <DT><STRONG><A href="#prepare" alt="Preparation">Before You Travel</A></STRONG>
    <DD><EM>How far can you go?<BR>
     What should you ware?<BR>
     What should you carry?<BR>
     What else should you do?</EM>
    </DL>
    <DL>
    <DT><STRONG><A href="#lost" alt="In To Trouble"> When Things Go Wrong</A></STRONG>
    <DD><EM>Stay!<BR>
      Think!<BR>
      Observe!<BR>
      Plan!</EM>
    </DL>
</TD>
<TD>
    <P><A name="intro">Walking in the woods,</A> whether as a hiker, camper,
    ➥backpacker or hunter, provides plenty of opportunities to practice the skills
    ➥of a backcountry traveler...</P>

    <BLOCKQUOTE><EM>This is in no way intended to be a course in backcountry safety
    ➥or survival. It's you against the world, and we're not taking any
    ➥responsibility for the outcome. </EM></BLOCKQUOTE>

    <H2><A name="prepare">Before you travel</A></H2>
    <H3>How far can you go? </H3>
    <P>When choosing a location and route, take into account your physical condition,
    ➥the reason for hiking, and the terrain and weather you could possibly
    ➥encounter...</P>

    <H3>What should you wear? </H3>
    <P>Proper foot gear is one of the most important parts of your journey. Sturdy
    ➥running shows are good for hiking on relatively smooth surfaces, such as
    ➥maintained trails...</P>

    <H3>What should you carry?</H3>
    <P>For short trips, a fanny pack or day pack should include at least one quart
    ➥of water per person, map, flashlight, first aid kit, rain gear, high-energy
    ➥snack, toilet paper...</P>

    <H2><A name="lost">When things go wrong</A></H2>

    <BOLCKQUOTE><EM>The worst thing you can do is to get frightened. The truly
     dangerous enemy is not the cold or the hunger, so much as the fear. It robs
     the wanderer of his judgement and of his limb power; it is fear that turns
     the passing experience into a final tragedy... Keep cool and all will be
     well... Use what you have, where you are,
     right now.</EM><BR>
    <CITE>Ernest Thompson Seton, 1906</CITE></BLOCKQUOTE>
    <P>So you've planned, you've prepared, and now you're on the trail. And the
     worst part is, you don't know where you are. There's a four step process to
     follow, and all you have to do is <EM>STOP</EM> --
     <STRONG>S</STRONG>tay-<STRONG>T</STRONG>hink- <STRONG>O</STRONG>bserve-
     <STRONG>P</STRONG>lan.<P>

  <EM>...More stuff here...</EM>

  </TD>
 </TR>
</TABLE>
```

Acest exemplu de tabel este un mod elementar de a realiza o pagină. Include un rând pentru antet și un rând pentru conținut. Dacă ați fi dorit să divizați în continuare conținutul dumneavoastră în secțiuni logice, puteați adăuga un rând suplimentar pentru fiecare parte – un rând pentru introducere, unul pentru pregătire, altul pentru situația în care utilizatorul se rătăcește și așa mai departe.

Pe măsură ce vă uitați prin listing, veți observa și altceva legat de conținutul din celule. Fiecare parte este marcată ca și cum nu ar fi parte din tabel. Etichetele de paragraf și alte elemente sunt folosite în continuare pentru a păstra formatul chiar dacă tabelul nu este redat de browserul utilizatorului. (Vezi Figura 10.2.)

Acum, că ați putut să vă faceți o idee despre ce este posibil de realizat, priviți un exemplu care este un pic mai avansat.

FIGURA 10.2.
Chiar și văzută pe un browser vechi, cum ar fi Mosaic 1.0, pagina își păstrează încă o mare parte din forma sa datorită etichetelor structurale utilizate împreună cu <TR> *și* <TD>.



Un corp în zigzag cu index

Următorul design este un pic mai interesant din punct de vedere vizual. La fiecare pauză pentru text pe oricare dintre cele două coloane, conținutul trece pe cealaltă coloană. (Vezi Figura 10.3.) Efectul de ferăstrău rezultat separă grafic fiecare secțiune de pe pagină, fără să întrerupă continuitatea textului în cadrul acelor secțiuni. Indexul care se desfășoară pe partea stângă furnizează o hartă fără să intervină în curgerea textului, iar o hiperlegătură în celulele nefolosite oferă o cale rapidă către index pentru navigare sus-jos.

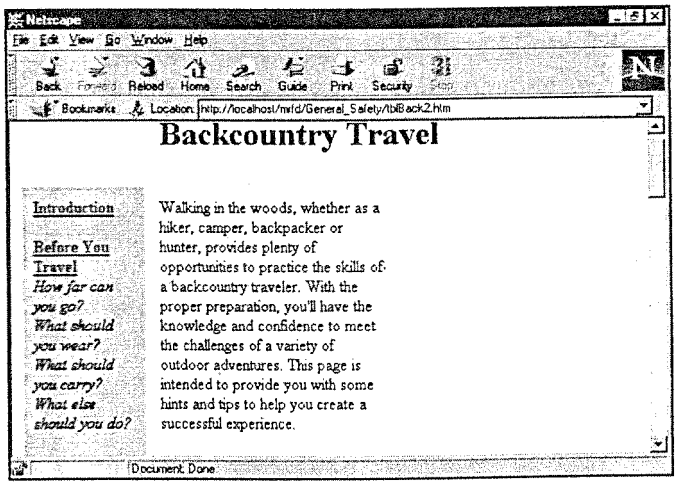
Cadrul de lucru pentru acest tabel include un rând pentru titlu, urmat de trei rânduri inițiale pentru conținut. Observați că sunt folosite atributele span pentru ca titlul să ocupe antetul celor două coloane, iar alt atribut span cuprinde rândurile indexului.

```
<TABLE border="0" width="100%" cellspacing="5" cellpadding="10">
  <TR>
    <TD>
      <!-- Empty cell to keep header from index column -->
    </TD>
    <TD colspan="2">
      <!-- HEADER CELL -->
```



```
    </TD>
    <!-- <TD< Unused cell from header span </TD> -->
  </TR>
  <TR>
    <TD width="16%" rowspan="3" valign="top" bgcolor="#C0C0C0">
      <!-- INDEX CELL -->
    </TD>
    <TD width="42%" valign="top">
      <!-- FIRST BODY CELL -->
    </TD>
    <TD width="42%" valign="bottom">
      <!-- Empty cell, ready for hyperlink at bottom -->
    </TD>
  </TR>
  <TR>
    <!-- <TD> Unused cell from index span </TD> -->
    <TD width="42%" valign="bottom">
      <!-- Empty cell, ready for hyperlink at bottom -->
    </TD>
    <TD width="42%" valign="top">
    <!-- SECOND BODY CELL -->
    </TD>
  </TR>
  <TR>
    <!-- <TD> Unused cell from index span </TD> -->
    <TD width="42%" valign="bottom">
      <!-- THIRD BODY CELL -->
    </TD>
    <TD width="42%" valign="top">
      <!-- Empty cell, ready for hyperlink at bottom -->
    </TD>
  </TR>
</TABLE>
```

FIGURA 10.3.
Conținutul alternează între cele două coloane la pauzele logice.



Din fericire, cadrul de lucru *doar pare* complicat. Comentariile din fiecare celulă vă arată de ce este nevoie în fiecare dintre ele. Primul rând este doar pentru titlu; el unește coloanele de corp. Al doilea rând începe cu celula de index și include apoi o celulă de conținut urmată de o celulă goală. Prima celulă de pe ultimul rând este absorbită de coloana de index, a doua este goală și a treia rămâne goală.

Poate vă întrebați în acest moment de ce coloana de index nu este plasată pe partea dreaptă a tabelului. Acest lucru se întâmplă deoarece pagina este redată pe un browser fără tabele. Cum browserul ignoră etichetele de tabel, primul conținut real pe care îl întâlnește este antetul și apoi indexul, ceea ce îl face pe acesta să apară primul în document, după cum se așteaptă și utilizatorul.

Dacă plasați indexul pe partea dreaptă, un utilizator care nu are capacitățile de tabel ar vedea antetul, apoi prima celulă de corp, urmată de index și de restul documentului – o ordine care l-ar zăpăci de tot pe bietul utilizator.

CÂND SĂ PLASAȚI COLOANA ÎNGUSTĂ ÎN DREAPTA

Există o situație în care plasați coloana de index în partea dreaptă – dacă ați transformat coloana într-un text suplimentar sau o notă. Atunci, cadrul de lucru elementar al tabelului ar arăta așa:

```
<TABLE border="0" width="100%" cellspacing="5" cellpadding="10">
 <TR>
  <TD colspan="2">
    <!-- HEADER CELL -->
  </TD>
  <!-- <TD< Unused cell from header span </TD> -->
  <TD>
    <!--Empty cell to keep header from sidebar column-->
  </TD>
 </TR>
 <TR>
  <TD width="42%" valign="top">
    <!-- FIRST BODY CELL -->
  </TD>
  <TD width="42%" valign="bottom">
    <!-- Empty cell, ready for hyperlink at bottom -->
  </TD>
  <TD width="16%" rowspan="2" valign="top" bgcolor="#C0C0C0">
    <!-- SIDEBAR CELL -->
  </TD>
 </TR>
 <TR>
  <TD width="42%" valign="bottom">
    <!-- Empty cell, ready for hyperlink at bottom -->
  </TD>
  <TD width="42%" valign="top">
    <!-- SECOND BODY CELL -->
  </TD>
  <!--<TD> Unused cell from index span </TD>-->
 </TR>
</TABLE>
```

Dacă acest cadru ar fi umplut cu conținut și apoi afișat pe un browser incompatibil, utilizatorul ar vedea antetul, apoi prima celulă de corp din conținut, urmată de textul suplimentar și apoi următoarea celulă de corp. Acest cadru menține o ordine logică în timp ce utilizatori vă citesc pagina.

Adăugarea conținutului unei machete de corp în zigzag cu index

După ce am studiat tabelul cu index și două coloane, este timpul să îl umplem cu conținut, după cum se vede în Listingul 10.2. Veți folosi conținutul din exemplul secțiunii anterioare.

Listingul 10.2. Adăugarea conținutului la tabelul cu index și două coloane.

```
<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 4.0//EN">
<TABLE border=0 width=100% cellspacing="5" cellpadding="10">
  <TR>
    <TD>
      <!-- Empty cell to keep header from index column -->
    </TD>
    <TD colspan="2">
      <!-- Header cell -->
      <H1>Backcountry Travel</H1>
    </TD>
    <!-- <TD> Unused cell from header span</TD> -->
  </TR>
  <TR>
    <TD width="20%" rowspan="3" valign="top" bgcolor="#C0C0C0">
      <!-- INDEX CELL -->
      <A name="index"></A>
      <P><STRONG><A href="#intro" alt="Introduction"> Introduction</A></STRONG></P>
      <P><STRONG><A href="#prepare" alt="Preparation"> Before You Travel</A>
      ➥</STRONG><BR>
        <EM>How far can you go?<BR>
        What should you wear?<BR>
        What should you carry?<BR>
        What else should you do?</EM></P>
      <P><STRONG><A href="#lost" alt="In To Trouble">When Things Go Wrong</A>
      ➥</STRONG><BR>
        <EM>Stay!<BR>
        Think!<BR>
        Observe!<BR>
        Plan!</EM>
    </TD>
    <TD width="40%" valign="top">
      <!-- FIRST BODY CELL -->
      <P><A name="intro">Walking in the woods,</A> whether as a hiker, camper,
      ➥backpacker or hunter, provides plenty of opportunities to practice the
      ➥skills of a backcountry traveler...</P>

      <P><EM>This is in no way intended to be a course in backcountry safety or
      survival. It's you against the world, and we're not taking any
      ➥responsibility for the outcome. </EM></P>
    </TD>
    <TD width="40%" valign="bottom">
      <!-- Empty cell, ready for hyperlink at bottom -->
      <A href="#index">Return to Index</A>
    </TD>
</TR>
<TR>
    <!-- <TD> Unused cell from index span </TD> -->
    <TD width="40%" valign="bottom">
```

(continuare)

Listingul 10.2. Continuare

```
        <!-- Empty cell ready for hyperlink at bottom -->
        <A href="#index">Return to Index</A>
    </TD>
    <TD width="40%" valign="top">
      <!-- SECOND BODY CELL -->
      <H2><A name="prepare">Before you travel</A></H2>
      <H3>How far can you go? </H3>
      <P>When choosing a location and route, take into account your physical
      ➥condition, the reason for hiking, and the terrain and weather you could
      ➥possibly encounter...</P>

      <H3>What should you wear? </H3>
      <P>Proper foot gear is one of the most important parts of your journey.
      ➥Sturdy running shows are good for hiking on relatively smooth surfaces, such
      ➥as maintained trails...</P>

      <H3>What should you carry?</H3>
      <P>For short trips, a fanny pack or day pack should include at least one
      ➥quart of water per person, map, flashlight, first aid kit, rain gear,
      ➥high-energy snack, toilet paper...</P>
    </TD>
  </TR
  <TR>
    <!-- <TD> Unused cell from index span </TD> -->
    <TD width="40%" valign="top">
      <!-- THIRD BODY CELL -->
      <H2><A name="lost">When things go wrong</A></H2>

      <P><EM>The worst thing you can do is to get frightened. The truly dangerous
       enemy is not the cold or the hunger, so much as the fear. It robs the
       wanderer of his judgement and of his limb power; it is fear that turns the
       passing experience into a final tragedy... Keep cool and all will be
       well... Use what you have, where you are, right now.</EM><BR>
      <CITE>Ernest Thompson Seton, 1906</CITE></P>

      <P>So you've planned, you've prepared, and now you're on the trail. And the
       worst part is, you don't know where you are. There's a four-step process
       to follow, and all you have to do is <EM>STOP</EM> --
      <STRONG>S</STRONG>tay-<STRONG>T</STRONG>hink-<STRONG>O</STRONG>bserve-
      <STRONG>P</STRONG>lan.<P>
    <EM>...More stuff here...</EM>
  </TD>
  <TD width="40%" valign="bottom">
      <!-- Empty cell, ready for hiperlink at bottom -->
      <A href="#index">Return to Index</A>
    </TD>
  </TR>
</TABLE>
```

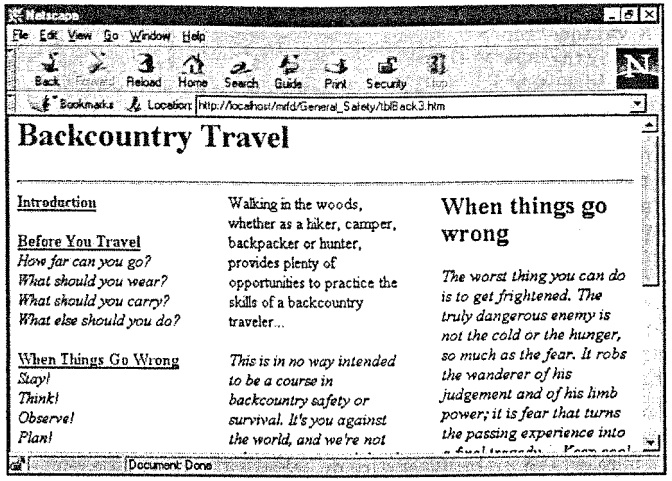
Așa cum este prezentat în acest tabel, conținutul a suferit câteva modificări. În primul rând, a fost înlăturată lista de definiții pentru index și au fost scoase citatele bloc pentru paragraful cu declinarea responsabilității și citatul din Seton. Coloanele fiind mai înguste, indenturile forțează textul în spații și mai strâmte, iar pagina pur și simplu nu arată bine.

Veți mai folosi acest conținut pentru încă un exemplu în acest capitol, ca să puteți vedea o metodă de a crea pagini interesante cu o unealtă simplă.

Macheta tradițională a unui ziar

Ultimul exemplu este puțin diferit față de cele două pe care le-ați văzut în acest capitol. Acesta creează o grilă care poate fi folosită într-o manieră mai tradițională pentru machetă. (Vezi Figura 10.4.)

FIGURA 10.4.
O machetă tradițională tip ziar folosește o grilă de celule.



Acest tip de machetă este realizat cu o structură de tabel destul de impresionantă. Setul de bază de etichete creează o grilă 5X5 – trei coloane de conținut și două coloane de margine (gutter) – la un loc cu un rând de antet. Când trebuie introdus conținutul și titlurile, veți folosi gruparea coloanelor și a rândurilor. Următoarea secțiune de cod creează un rând de antet, urmat de cinci rânduri împărțite în cinci celule fiecare:

```
<TABLE border="0" width="100%" cellspacing="0" cellpadding="0">
 <TR>
  <TD colspan="5">
    <!-- HEADER CELL -->
  </TD>
  <!-- 4 unused <TD> cells here -->
 </TR>
 <TR valign="top">
  <TD width="30%"> </TD>
  <TD width="5%">
  <TD width="30%"> </TD>
  <TD width="5%">
  <TD width="30%"> </TD>
 </TR>
 <TR valign="top">
  <TD width="30%"> </TD>
  <TD width="5%">
  <TD width="30%"> </TD>
  <TD width="5%">
  <TD width="30%"> </TD>
 </TR>
```

```
 <TR valign="top">
  <TD width="30%"> </TD>
  <TD width="5%">
  <TD width="30%"> </TD>
  <TD width="5%">
  <TD width="30%"> </TD>
 </TR>
 <TR valign="top">
  <TD width="30%"> </TD>
  <TD width="5%">
  <TD width="30%"> </TD>
  <TD width="5%">
  <TD width="30%"> </TD>
 </TR>
 <TR valign="top">
  <TD width="30%"> </TD>
  <TD width="5%">
  <TD width="30%"> </TD>
  <TD width="5%">
  <TD width="30%"> </TD>
 </TR>
</TABLE>
```

Acest tip de machetă este realizat cel mai bine într-un editor cu o interfață vizuală, astfel încât să puteți vedea rezultatele pe măsură ce îl editați. Dezavantajul major al acestei machete este că un browser care nu are suport pentru tabele nu o redă întotdeauna destul de bine. Dacă aveți de gând să folosiți acest stil, este o idee bună să furnizați o versiune tradițională, care să prezinte materialul în ordinea corectă.

SUGESTIE

Cea mai ușoară metodă să aflați cum este prezentat un tabel pe un browser care nu are suport pentru tabele este să citiți codul HTML consecutiv. Fiecare set de conținut apare pe o pagină Web în ordinea în care îl întâlnește browserul. Cu alte cuvinte, celulele din fiecare rând sunt redate de la stânga la dreapta, în ordinea în care sunt reprezentate rândurile.

Adăugarea conținutului unei machete tip ziar

Celula titlului este destul de ușor de înțeles, așa că începeți cu prima porțiune de conținut – indexul. Această parte a textului ar trebui să ocupe partea de stânga-sus a machetei, deci puneți-o pe primul rând, după antet, în prima celulă:

```
<TD width="30%">
  <P><STRONG><A href="#intro" alt="Introduction"> Introduction</A></STRONG></P>
  <P><STRONG><A href="#prepare" alt="Preparation"> Before You Travel</A>
    ➥</STRONG><RR>
   <EM>How far can you go?<BR>
     What should you wear?<BR>
     What should you carry?<BR>
     What else should you do?</EM></P>
  <P><STRONG><A href="#lost" alt="In To Trouble"> When Things Go Wrong</A>
  ➥</STRONG><BR>
   <EM>Stay!<BR>
   Think!<BR>
```



```
   Observe!<BR>
   Plan!</EM>
</TD>
<TD width="5%">
```

Apoi, adăugați introducerea celulei alăturate:

```
<TD width="30%">
  <P><A name="intro">Walking in the woods,</A> whether as a hiker, camper,
    ➥backpacker or hunter, provides plenty of opportunities to practice the
    ➥skills of a backcountry traveler...</P>
  <P><EM>This is in no way intended to be a course in backcountry safety or
    ➥survival. It's you against the world, and we're not taking any
    ➥responsibility for the outcome. </EM></P>
</TD>
<TD width="5%">
```

Informațiile despre ce să faci dacă te rătăcești ocupă toată coloana a treia, așa că adăugați-le folosind atributul `rowspan`. Trebuie să interveniți, de asemenea, și în celelalte rânduri și să vă asigurați că ultima celulă din fiecare rând este ștearsă sau transformată în comentariu, astfel încât să nu afecteze afișarea tabelului:

```
<TD WIDTH="30%" ROWSPAN="5">
  <H2><A name="lost">When things go wrong</A></H2>
  <P><EM>The worst thing you can do is to get frightened. The truly dangerous
   enemy is not the cold or the hunger, so much as the fear. It robs the wanderer
   of his judgement and of his limb power; it is fear that turns the passing
   experience into a final tragedy... Keep cool and all will be well... Use what
   you have, where you are, right now.</EM><BR>
   <CITE>Ernest Thompson Seton, 1906</CITE></P>
  <P>So you've planned, you've prepared, and now you're on the trail. And the
   worst part is, you don't know where you are. There's a four step process to
   follow, and all you have to do is <EM>STOP</EM>--
  <STRONG>S</STRONG>tay-<STRONG>T</STRONG>hink-<STRONG>O</STRONG>bserve-
   <STRONG>P</STRONG>lan.<P>
  <EM>...More stuff here...</EM>
</TD>
```

Acum sunteți gata să începeți rândul următor, care conține titlul corpului de text despre cum să ne pregătim. Ar trebui să se extindă peste tot articolul (două coloane), deci include atributul `colspan` pentru a combina prima și a doua coloană cu gutter-ul. Este adăugată o linie orizontală pentru a separa antetul de conținutul care îl precede, iar atributul `valign` este stabilit la valoarea `bottom` pentru a forța titlul mai aproape de text. Deoarece ultima coloană este ocupată cu corpul de text din rândul precedent, este deja transformat în comentariu:

```
<TD width="30%" colspan="3" valign="bottom">
 <HR>
 <H2><A name="prepare">Before you travel</A></H2>
</TD>
<!-- unused cells <TD width="5%"><TD width="30%"> -->
<TD width="5%">
<!-- <TD width="30%"> </TD> -->
```

În următorul rând sunt plasate două coloane de conținut pentru antet, la un loc cu altă etichetă comentată pentru a treia coloană:

```
<TD width="30%">
  <P><STRONG>How far can you go? </STRONG>
```

```
    When choosing a location and route, take into account your physical condition,
    ➥the reason for hiking, and the terrain and weather you could possibly
    ➥encounter...</P>
</TD>
<TD width="5%">
<TD width="30%">
  <P><STRONG>What should you wear? </STRONG>
   Proper foot gear is one of the most important parts of your journey. Sturdy
   ➥running shows are good for hiking on relatively smooth surfaces, such as
   ➥maintained trails...</P>
</TD>
<TD width="5%">
<!-- <TD width="30%"> </TD> -->
```

Ultimele două rânduri rămân la fel în caz că veți avea nevoie să extindeți conținutul ulterior. Ultimele două celule din aceste rânduri sunt date la o parte pentru a face loc pentru a treia coloană de text, care se întinde către partea de jos a tabelului.

Tabelul finisat este arătat în Listingul 10.3. Nu este cel mai ușor sau cel mai arătos tabel care poate fi asamblat, dar, executat corect, poate să imite înfățișarea unui ziar și, în același timp, să ofere un conținut la subiect, putând fi interpretat de cât mai multe browsere posibile.

Listingul 10.3. Macheta completă de ziar.

```
<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 4.0//EN">
<TABLE border=0 width=100% cellspacing="0" cellpadding="0" valign="top">
 <TR>
  <TD colspan="5">
    <!-- HEADER CELL -->
    <H1>Backcountry Travel</H1>
    <HR>
  </TD>
  <!-- 4 unused <TD> cells here -->
 </TR>
 <TR valign="top">
  <TD width="30%">
    <P><STRONG><A href="#intro" alt="Introduction"> Introduction</A></STRONG></P>
    <P><STRONG><A href="#prepare" alt="Preparation"> Before You Travel</A>
    ➥</STRONG><BR>
     <EM>How far can you go?<BR>
     What should you wear?<BR>
     What should you carry?<BR>
     What else should you do?</EM></P>
    <P><STRONG><A href="#lost" alt="In To Trouble"> When Things Go Wrong</A>
    ➥</STRONG><BR>
     <EM>Stay!<BR>
     Think!<BR>
     Observe!<BR>
     Plan!</EM>
  </TD>
  <TD width="5%">
  <TD width="30%">
```



```
    <P><A name="intro">Walking in the woods,</A> whether as a hiker, camper,
     ➥backpacker or hunter, provides plenty of opportunities to practice the
     ➥skills of a backcountry traveler...</P>
    <P><EM>This is in no way intended to be a course in backcountry safety or
     ➥survival. It's you against the world, and we're not taking any
     ➥responsibility for the outcome. </EM></P>
  </TD>
  <TD width="5%">
  <TD WIDTH="30%" ROWSPAN="5">
    <H2><A name="lost">When things go wrong</A></H2>
    <P><EM>The worst thing you can do is to get frightened. The truly dangerous
     enemy is not the cold or the hunger, so much as the fear. It robs the
     wanderer of his judgement and of his limb power; it is fear that turns the
     passing experience into a final tragedy... Keep cool and all will be well...
     Use what you have, where you are, right now.</EM><BR>
<CITE>Ernest Thompson Seton, 1906</CITE></P>
    <P>So you've planned, you've prepared, and now you're on the trail. And the
     worst part is, you don't know where you are. There's a four step process to
     follow, and all you have to do is <EM>STOP</EM>--
    <STRONG>S</STRONG>tay-<STRONG>T</STRONG>hink-<STRONG>O</STRONG>bserve-
    <STRONG>P</STRONG>lan.<P>
    <EM>...More stuff here...</EM>
  </TD>
</TR>
<TR valign="top">
  <TD width="30%" colspan="3" valign="bottom">
    <HR>
    <H2><A name="prepare">Before you travel</A></H2>
  </TD>
  <!-- unused cells <TD width="5%"><TD width="30%"> -->
  <TD width="5%">
  <!-- <TD width="30%"> </TD> -->
</TR>
<TR valign="top">
  <TD width="30%">
  <P><STRONG>How far can you go? </STRONG>
   ➥When choosing a location and route, take into account your physical
   ➥condition, the reason for hiking, and the terrain and weather you could
   ➥possibly encounter...</P>
  </TD>
  <TD width="5%">
  <TD width="30%">
  <P><STRONG>What should you wear? </STRONG>
   Proper foot gear is one of the most important parts of your journey. Sturdy
   ➥running shows are good for hiking on relatively smooth surfaces, such as
   ➥maintained trails...</P>
  </TD>
  <TD width="5%">
  <!-- <TD width="30%"> </TD> -->
</TR>
<TR valign="top">
    <TD width="30%"> </TD>
    <TD width="5%">
    <TD width="30%"> </TD>
    <TD width="5%">
    <!-- <TD width="30%"> </TD> -->
```

(continuare)

Listingul 10.3. Continuare

```
</TR>
  <TR valign="top">
    <TD width="30%"> </TD>
    <TD width="5%">
    <TD width="30%"> </TD>
    <TD width="5%">
    <!-- <TD width="30%"> </TD> -->
  </TR>
</TABLE>.
```

Rezumat

Tabelele au fost inițial proiectate pentru afișarea simplă a informațiilor și datelor într-o formă tabulară și ordonată (vezi Capitolul 9, "Organizarea datelor cu ajutorul tabelelor"). Dar, aceeași flexibilitate care permite afișarea eficientă a datelor permite tabelelor să creeze o mulțime de machete diferite de pagini.

Ați văzut trei exemple de machete în acest capitol, iar în Partea a VII-a, „Designul efectiv al unei pagini Web", veți găsi și alte idei de utilizare a tabelelor și a foilor de stil. Acestea ar trebui să vă facă să creați pagini care să depășească afișarea prestabilită a unei pagini Web.

Cu aceste capacități, totuși, apar și noi responsabilități. Amintiți-vă că nu toate browserele au suport pentru tabele – ceea ce înseamnă că este de datoria dumneavoastră să vă asigurați că tabelele vor funcționa corespunzător pe diverse browsere și platforme, indiferent dacă browserul utilizatorului este activat pentru tabele sau nu.

Când pagina dumneavoastră este afișată pe un browser fără tabele, browserul afișează informațiile în aceeași ordine în care le-a găsit în etichetele de tabel. Dacă nu ați inclus etichete container (paragrafe, diviziuni sau orice altă formatare la nivelul blocului) pentru conținutul celulelor dumneavoastră, atunci se vor uni toate într-un singur paragraf.

Dacă țineți minte aceste lucruri pe măsură ce veți lucra la propriile dumneavoastră soluții creative de machete, veți produce pagini Web atractive și, în același timp, fără probleme de redare.

Capitolul 11
Adăugarea imaginilor pe pagina dumneavoastră Web

de Rick Darnell

În acest capitol

- **Eticheta <IMG> de bază 147**
- **Includerea unei imagini ca <OBJECT> 153**
- **Folosirea imaginilor pe post de înlocuitor al conținutului 154**

Imaginile sunt la fel de des întâlnite în paginile Web ca și în reviste. Este adevărat că există unele situri care nu includ imagini, dar acestea sunt foarte puține. Chiar și siturile în care abundă textul conțin imaginea emblemei companiei sau un alt simbol grafic.

Oricât de răspândite ar fi, imaginile sunt o caracteristică relativ nouă a Webului. Rădăcinile Webului sunt domeniul apărării și cel academic, care nu prea necesitau poze pe calculator. Dar, pe măsură ce Webul s-a extins în domeniul utilizatorului obișnuit, creatorii de pagini au dorit să posede abilitatea de a include grafică.

De acest lucru s-au ocupat cei de la National Center for Supercomputing Applications (NCSA), o organizație a Universității Illinois. Ei au asamblat un browser numit Mosaic, care avea suport pentru afișarea imaginilor chiar pe pagină, la un loc cu restul textului. Mosaic, a fost primul browser grafic și baza pentru aproape toate browserele create de atunci încoace, inclusiv pentru Netscape Navigator și Microsoft Internet Explorer.

NCSA a realizat adăugarea graficii la HTML prin introducerea unei etichete numite <IMG>. Această etichetă a permis autorilor să introducă o poză în același rând cu restul textului. Această etichetă mai există încă și permite câteva trucuri noi-nouțe, pe care vi le voi arăta în acest capitol. Tot aici vă voi prezenta câțiva prieteni și câteva rude ale etichetei, care sunt asociați cu imaginile. Această caracteristică a parcurs un drum lung de la crearea sa, iar dumneavoastră veți învăța o multitudine de metode pentru a ajusta mai eficient înfățișarea și poziționarea.

Deși eticheta <IMG> a devenit în timp unul dintre pilonii de rezistență ai HTML-ului, se pare că și aceasta a ajuns în cele din urmă la capătul carierei sale. Eticheta <OBJECT>, introdusă în HTML 4, include capacitatea de a adăuga imagini, plug-in-uri, miniaplicații și alte tipuri de medii, la un loc cu posibilități extinse pentru conținut opțional, când acesta nu este acceptat sau este afișat pentru alte scopuri.

Eticheta <IMG> de bază

Eticheta <IMG> este un element gol, folosit la introducerea imaginilor inline, începând cu mici pictograme și simboluri grafice și până la hărți imagine care ocupă aproape toată fereastra browserului. Deoarece această etichetă este un element pentru o singură resursă (o etichetă pentru o imagine), nu este prevăzută o etichetă de închidere.

În plus față de identificarea imaginii ce urmează a fi folosită, eticheta are și atribute pentru a defini poziția acesteia în raport cu textul înconjurător și conținutul Web. Aceste poziții includ mutarea imaginii în marginea din stânga sau dreapta sau plasarea sa deasupra, dedesubt sau centrat pe linia de text pe care apare.

Sintaxa pentru o etichetă imagine este:

```
<IMG src=URL [alt=textDescription][core][events][alignment]>
```

Atributul `src` identifică numele fișierului imagine (*filename*) cu o adresă URL și un nume de fișier fizice sau relative. Atributul `alt` definește o scurtă descriere de tip text a imaginii sau a scopului său pentru browserele care nu o încarcă. Atributele *core* și *events* sunt explicate în textul suplimentar din introducerea acestei cărți.

SUGESTIE

Browserele pot să nu încarce imaginile dintr-o multitudine de motive. În primul rând, browserul ar putea fi nongrafic, cum este Lynx. În al doilea rând, utilizatorul ar putea avea încărcarea automată a imaginilor dezactivată pentru a iuți timpul de încărcare. Furnizând o valoare lui `alt`, vă asigurați că utilizatorul va avea o idee despre ce ar fi trebuit să umple acel mare spațiu gol.

Atributul `alt` este vital pentru interoperabilitatea cu agenți de utilizator bazați doar pe text sau vorbire. Pentru cei care nu au posibilitatea de a vedea imaginea, atributul `alt` poate oferi o scurtă descriere a acesteia. În cazul browserelor care nu redau decât textul, este singura indicație că utilizatorul pierde o bucată din conținut.

Atributul `src` este necesar pentru fiecare imagine. El identifică ce imagine să fie folosită și tipul acesteia. Cele mai folosite două tipuri de imagini sunt GIF (Graphics Interchange Format) și JPEG (Joint Photographic Experts Group, folosit și ca JPG), dar și imaginile PNG (Portable Network Graphic) sunt din ce în ce mai acceptate și mai răspândite.

Al doilea atribut din definiția sintaxei, `alt`, este opțional, dar recomandat. Furnizează o descriere de tip text a imaginii și este singura parte din eticheta `<IMG>` folosită de browserele care nu acceptă imaginile inline.

UNDE SĂ ALINIAȚI IMAGINEA

Ca multe alte atribute de înfățișare, atributul `align` este nerecomandat în HTML 4.0, fiind preferat controlul aspectului prin intermediul foilor de stil. Deocamdată `align` este încă folosit, pentru compatibilitatea cu browserele mai vechi, dar este pe drumul spre ieșire. O discuție mai amplă despre `align` puteți găsi în Capitolul 17, „Stil simplu cu text". Ca o scurtă trecere în revistă, iată cum poziționează `align` imaginile față de linia textului pe care apar. Spre deosebire de alte atribute de aliniere a articolelor, cum ar fi cele pentru tabele, `align` controlează atât plasamentul vertical, cât și cel orizontal. Iată sintaxa sa:

```
<IMG src="URL" align="position">
```

În această linie, *position* este una dintre următoarele cinci valori: `top`, `middle`, `bottom`, `left` sau `right`. Efectul fiecăreia dintre aceste valori este după cum urmează:

- **`Top`:** Poziționează partea superioară a imaginii în partea superioară a rândului curent. Dacă rândul de text este formatat cu o etichetă specifică, cum ar fi `<H1>`, imaginea pare să ocupe mai mult spațiu în rând decât dacă ar apărea pe o linie a unui corp de text standard. Browserele nu se comportă la fel; unele folosesc rândul anterior imaginii pentru a determina alinierea acesteia, iar altele îl folosesc pe cel care urmează.
- **`Middle`:** Asemănător cu `top`, dar înălțimea mediană a imaginii este aliniată cu linia de bază a rândului curent. (Vezi Figura 11.1.) Interpretarea sa nu variază în funcție de browser.
- **`Bottom`:** Este valoarea prestabilită atunci când `align` nu este specificat în eticheta `<IMG>`. În acest caz, partea de jos a imaginii se sprijină pe linia de bază a rândului curent de text.
- **`Left`:** Forțează imaginea către marginea stângă curentă și orice text care o urmează apare în dreptul marginii din dreapta a acesteia. Interpretarea sa variază dacă alte imagini sau materiale au apărut înaintea ei aliniate la stânga. Textul precedent imaginii o forțează de obicei către un rând nou, iar textul care îi urmează continuă pe rândul precedent imaginii.

- **`Right`:** Are un efect asemănător cu cel al valorii `left`, dar imaginea este forțată către marginea din dreapta. Textul care îi urmează învăluie partea stângă a imaginii. Comportamentul său este același cu al lui `left` dar în direcția contrară, variind în funcție de textul și materialele precedente. (Vezi Figura 11.2.)

Unele browsere introduc spațieri suplimentare ale rândurilor cu mai multe imagini care folosesc alinierile `left` sau `right`, dar să nu vă așteptați ca spațierea să fie aceeași pe toate browserele și platformele. Pentru mai multe informații despre controlarea curgerii textului, vezi Capitolul 7, „Alinierea și formatarea textului".

Când plasați o imagine pe o pagină, amintiți-vă că aceasta este o caracteristică inline, afișată la un loc cu textul de pe același rând. Dacă nu este separată de materialul înconjurător cu o întrerupere de rând sau de paragraf, imaginea va fi plasată pe aceeași linie cu rândul curent de text, conform atributului `align`. Acest plasament poate conduce la anumite înfățișări nu tocmai dorite ale paginii, după cum se poate vedea în Figura 11.3.

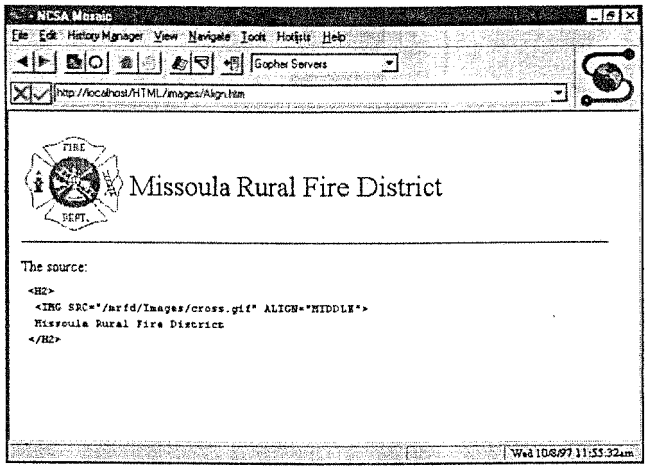


FIGURA 11.1. *Textul pentru pictogramă este în linie cu centrul imaginii.*

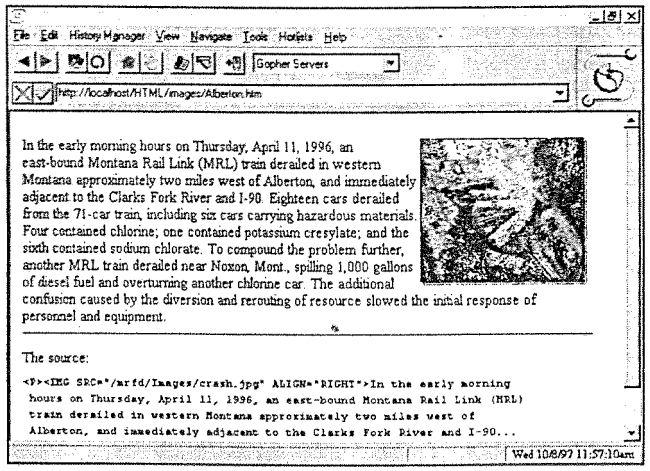
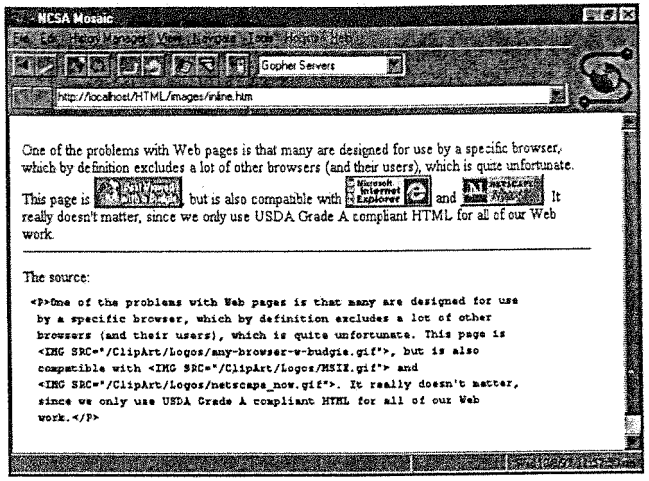


FIGURA 11.2. *Imaginea este aliniată la dreapta, cu textul care îi urmează scris în stânga sa.*

FIGURA 11.3. *Deși imaginile sunt mici, tot afectează spațierea rândurilor, dând paragrafului un aspect puțin mai ciudat.*



Crearea spațiului cu atributele `width` și `height`

Cele două atribute ale mărimii imaginii – `width` și `height` – setează spațiul orizontal și vertical pentru imagine în pixeli. Ele sunt de obicei folosite ca pereche pentru a rezerva spațiu în fereastra browserului înainte ca imaginea să fie încărcată. Sintaxa este:

```
<UMG src="URL" width=pixels height=pixels>
```

URL este numele imaginii, iar *pixels* este mărimea spațiului rezervat imaginii.

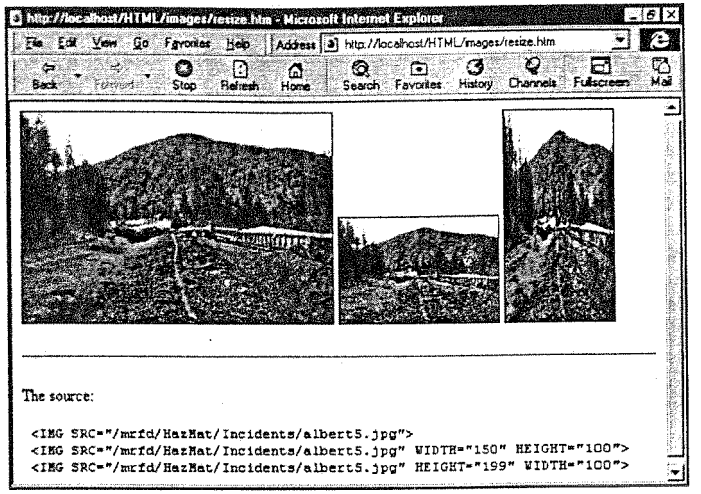
Pentru a obține rezultate consecvente și previzibile, imaginile ar trebui folosite în mărimea lor reală. Totuși, acest lucru nu este posibil întotdeauna, din diferite motive. În majoritatea browserelor, atributele `width` și `height` au prioritate față de dimensiunea reală a imaginii, dându-vă o modalitate ușoară de a forța imaginea într-un spațiu diferit ca mărime față de aceasta.

Totuși, această metodă de a schimba mărimea imaginii are unele dezavantaje. Primul, dacă o imagine mică este mărită, calitatea acesteia are de suferit. Dacă o imagine mare este micșorată, timpul de descărcare este mărit datorită unui fișier mai mare decât este nevoie. Dacă valorile specificate în eticheta `<IMG>` nu mențin proporțiile înălțime-lățime ale originalului, imaginea va apărea distorsionată pe o pagină Web, după cum se poate vedea în Figura 11.4.

Există două motive pentru care să doriți să specificați mărimea imaginii dumneavoastră. În primul rând, așa cum deja am menționat, puteți reduce timpul de afișare pentru restul paginii. Când informațiile despre mărime sunt lăsate la o parte, unele browsere rezervă un spațiu minim și apoi încep descărcarea primei bucăți a imaginii, care include și informațiile sale despre mărime. În timp ce browserul lucrează la acest lucru, nu mai descarcă și altceva din pagină.

În funcție de setările browserului, restul paginii s-ar putea să nu fie afișat până când acesta nu va ști de cât spațiu are nevoie fiecare imagine sau afișajul s-ar putea actualiza și reîncărca pe măsură ce primește fiecare informație. Folosirea atributelor `width` și `height` înlătură munca de „ghicire" din partea browserului.

FIGURA 11.4. *Prima imagine este în forma sa originală; celelalte două sunt modificate prin atributele lor* `width` *și* `height`.



Cele două atribute sunt de asemenea folosite pentru a păstra formatarea paginii. Dacă aspectul total al paginii dumneavoastră depinde de mărimile și relațiile dintre imagini și text (ca în cazul unui ziar sau al unei reviste), specificarea mărimi imaginii asigură rezervarea spațiului necesar. Chiar dacă imaginea nu va apărea, spațiul necesar ei este păstrat liber, pentru a asigura efectul pe care îl doriți.

Ați dori și un chenar la respectiva imagine?

Când o imagine apare ca parte dintr-o legătură de hipertext, browserul de obicei răspunde desenând în jurul imaginii un chenar colorat (în mod obișnuit, albastru). Lățimea acestui chenar este fixată prin folosirea atributului `border`. Sintaxa este:

```
<IMG src="URL" border=pixels>
```

URL este calea și numele fișierului imaginii, iar *pixels* este lățimea chenarului în pixeli. Folosiți valoarea `0` pentru a ascunde chenarul. Culoarea chenarului este controlată prin atributele pentru culoarea legăturii din foaia de stil sau din eticheta `<BODY>`.

Majoritatea browserelor indică de asemenea o imagine de hiperlegătură prin schimbarea indicatorului mouse-ului când acesta trece peste grafică, astfel încât utilizatorul nu trebuie să ghicească dacă o imagine fără chenar este o hiperlegătură.

Dați imaginii un pic de `hspace` și `vspace`

Atributele de spațiu creează o *zonă tampon* în jurul perimetrului imaginii, care este foarte folositoare când sunt necesare spații albe în imediata apropiere a imaginii. Atributele `hspace` și `vspace` fixează mărimea acestor spații albe în pixeli. Sintaxa este:

```
<IMG src="URL" hspace=pixels vspace=pixels>
```

URL este fișierul imagine, iar *pixels* este numărul de pixeli adăugat marginii corespunzătoare. În mod prestabilit, amândouă atributele de spațiu au ca valori numere mici, diferite de zero, care furnizează exact atât spațiu alb cât este necesar pentru ca imaginea să nu atingă textul adiacent.

OBSERVAȚIE

Spațiu alb este un termen din design pentru „spațiu care nu conține nimic". Nu înseamnă neapărat că este alb. În funcție de culoarea de fundal a paginii, spațiul alb ar putea fi și verde, albastru, roz sau orice altă culoare.

Spațiul este adăugat în amândouă părțile atributului. Așadar, dacă includeți valoarea 40 pentru `hspace`, sunt adăugați 40 de pixeli la stânga și la dreapta imaginii. Însă această cantitate de spațiu face ca imaginea să nu pară a fi aliniată cu celelalte margini ale paginii. (Vezi Figura 11.5.)

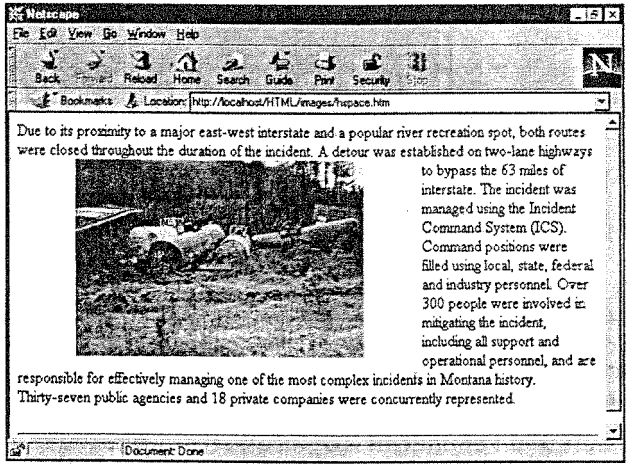


FIGURA 11.5. *Spațiul orizontal din jurul imaginii creează impresia unor margini inegale față de textul înconjurător.*

Atributul `ismap` pentru hărțile server-side

Atributul `ismap` identifică imaginea ca pe o hartă imagine, folosită pentru a asocia anume părți din imagine cu hiperlegături către alte documente.

Pentru a fi valabilă, o etichetă `<IMG>` cu `ismap` este închisă într-o etichetă de ancoră de hipertext. Când utilizatorul execută un clic pe imagine, atributul `ismap` transmite coordonatele x și y ale locului unde s-a întâmplat acest lucru către server. Sintaxa este:

```
<a href="URL/file.map"><img src="imageURL" ismap></a>
```

`URL/file` este numele fișierului care conține coordonatele hărții imagine (folosind o extensie `.map`), iar *`imageURL`* este numele imaginii propriu-zise. Aceasta se numește hartă imagine *server-side*, deoarece informațiile hiperlegăturii sunt procesate de un server.

SUGESTIE

Atributul `ismap` este folosit numai cu hărți imagine sever-side. Serverul trebuie să dețină programele CGI corespunzătoare pentru a procesa informațiile hărții imagine și fișierul care definește fiecare punct fierbinte.

Locul din imagine în care utilizatorul execută clic este transmis serverului prin crearea unei noi adrese URL din cea specificată în eticheta de ancoră. Este adăugat un semn de întrebare la sfârșit, urmat de valoarile coordonatelor x și y separate de o virgulă. Apoi este urmată legătura, folosind noul URL. Ca exemplu, dacă utilizatorul execută clic la locația `x=10` și `y=27`, adresa URL trimisă serverului este `/cgibin/navbar.map?10,27`.

SUGESTIE

Este o idee bună să folosiți `border=0` pentru imagine și să introduceți indicii grafice pentru a informa utilizatorul că imaginea este o hartă pe care poate executa clic. Altfel, este ușor de confundat o hartă imagine cu o simplă imagine de hiperlegătură. Indiciile grafice includ tehnici cum ar fi forme îngroșate cu text descriptiv sau o formă orizontală lungă în macheta tradițională de bară de navigare.

O caracteristică recent adăugată browserelor este capacitatea de a interpreta o hartă imagine fără nici un ajutor din partea serverului. Numite *hărți imagine client-side*, ele includ toate informațiile necesare pentru a-și procesa propriile hiperlegături fără a mai comunica cu serverul. Acest tip de hartă imagine este realizată cu atributul `usemap`. Puteți găsi mai multe informații despre hărți imagine în Capitolul 14, „Crearea meniurilor cu hărți imagine".

Usemap pentru hărți imagine client-side

Atributul `usemap` marchează o imagine ca fiind o hartă imagine client-side care este folosită cu o etichetă `<MAP>`. În majoritatea cazurilor, o hartă client-side este preferată față de o hartă server-side, deoarece nu necesită comunicarea pe liniile rețelei și nici un suport adițional din partea serverului pentru a funcționa. Sintaxa este următoarea:

```
<IMG src="URL" usemap="[mapURL.html]#mapName">
```

`URL` este fișierul imagine, iar *`[mapURL.html]#mapName`* este o adresă URL incompletă care specifică locația elementului `<MAP>` cu informațiile coordonatelor și hiperlegăturii și care este asemănător cu o ancoră de țintă. Dacă nu este specificat numele fișierului, se presupune că elementul `<MAP>` există în fișierul HTML curent.

OBSERVAȚIE

Observați diferența dintre extensia fișierului specificat pentru ancora hiperlegăturii în `ismap` și sursa coordonatelor pentru `usemap`. Prima necesită un fișier MAP, care constă într-o colecție de coordonate și hiperlegături. Ultima este un bloc de etichete HTML dintr-un fișier HTML. Fișierul poate conține și altceva sau poate exista doar ca să depoziteze informațiile hărții.

Regiunile active ale hărții imagine sunt descrise browserului prin folosirea etichetelor `<MAP>` și `<AREA>`. Această informație este de obicei plasată în cadrul fișierului HTML împreună cu eticheta `<IMG>` care o solicită, dar poate fi plasată și într-un fișier separat, dacă aceeași hartă este folosită în mai multe pagini.

Dacă doriți să asigurați compatibilitatea cu mai multe browsere și platforme, pentru aceeași imagine puteți folosi și `ismap`, și `usemap`, ceea ce permite browserului să aleagă în ce fel vrea să interpreteze harta imagine. Mai multe informații despre aceste două atribute se află în Capitolul 14.

Includerea unei imagini ca `<OBJECT>`

Odată domeniul exclusiv al controalelor Microsoft ActiveX, eticheta `<OBJECT>` a fost adoptată de Consorțiul World Wide Web ca o unealtă de uz general pentru a insera orice tip de conținut pe o pagină Web, inclusiv miniaplicații, plug-in-uri, controale înglobate și, bineînțeles, imagini. Puteți citi mai multe despre toate aceste întrebuințări în Capitolul 16, „Obiecte înglobate în HTML", dar acum voi discuta doar despre imagini.

Aceasta este cea mai simplă sintaxă pentru adăugarea unei imagini cu eticheta `<OBJECT>`:

```
<OBJECT src="imgURL">alternateContent</OBJECT>
```

imgURL este adresa URL a fișierului imagine. În loc să folosiți atributul `alt` pentru conținutul alternativ, ca în eticheta autonomă `<IMG>`, eticheta `<OBJECT>` vă permite să imbricați aproape orice conținut HTML în cadrul granițelor sale în vederea afișării de către orice browser incompatibil. De exemplu, următorul cod afișează textul încastrat și formatarea sa dacă browserul nu recunoaște eticheta `<OBJECT>`:

```
<OBJECT src="festus.gif">
A picture of <STRONG>Festus</STRONG>,
black panther of the north and <EM>my cat</EM>.
</OBJECT>
```

PERICOL!

Din păcate, această întrebuințare a etichetei `<OBJECT>` nu este încă acceptată de nici unul dintre principalele browsere. Navigator va ignora etichetele și va afișa în mod prestabilit conținutul alternativ, în care ar trebui să includeți eticheta `<IMG>`. Internet Explorer se va bloca dacă nu întâlnește fișierul său `.OCX` și va afișa pictograma care semnalează un conținut necorespunzător (un desen care seamănă cu o pagină cu un X roșu), pe care nu o puteți ocoli.

Fiți atent la Internet Explorer – când va include suport complet pentru eticheta `<OBJECT>`, puteți începe să folosiți această tehnică pentru introducerea imaginilor. Între timp, aceasta este o informație folositoare pentru viitor.

Există mai multe atribute pe care le puteți folosi cu imaginile introduse în acest mod. Pentru formate noi de imagini care nu sunt acceptate în mod obligatoriu de toate browserele, cum ar fi PNG, puteți specifica tipul MIME. Această metodă dă browserului opțiunea să ocolească încărcarea conținutului și să se mute la textul alternativ, după cum urmează:

```
<OBJECT src="festus.png" type="image/x-png">
  <OBJECT src="festus.gif" type="image.gif" standby"=Loading large image of my
  ➥cat.">
    A picture of <STRONG>Festus</STRONG>,
    black panther of the north and <EM>my cat</EM>.
  </OBJECT>
</OBJECT>
```

Aici, eticheta de la margine specifică o imagine PNG. Dacă browserul decide că nu poate manevra acest tip de imagine, poate ignora eticheta și trece la următoarea, care include o imagine GIF. În acest caz, ați presupus că este un GIF mare și ineficient care va dura mult timp să fie încărcat, așa că ați furnizat un mesaj `standby` (de așteptare) pentru a fi afișat în spațiul imaginii până când fișierul se încarcă din server. În cele din urmă, dacă utilizatorul are un browser bazat pe text care nu acceptă imaginile, este furnizat un text alternativ.

Metoda `<OBJECT>` vă conferă o flexibilitate sporită pentru includerea imaginilor pe pagina dumneavoastră și se va dovedi mult mai folositoare pe măsură ce progresează Webul.

Folosirea imaginilor pe post de înlocuitor al conținutului

Având în vedere marea varietate de browsere și capacități de pe Web, este din ce în ce mai greu să creezi o pagină care să funcționeze pentru toți. Acest lucru este în special adevărat când includeți conținuturi cum ar fi miniaplicații Java, controale ActiveX și plug-in-uri Netscape. Dacă un browser nu recunoaște conținutul dumneavoastră special, îl ignoră și lasă o gaură sau un spațiu alb și mare în machetă.

Puteți folosi incompatibilitatea browser-etichetă în avantajul dumneavoastră, prin plasarea unei etichete de imagine imediat înainte de eticheta de închidere a conținutului special. Iată cum arată formatul:

```
<APPLET attributes height=contentHeight width=contentWidth>
<PARAM attributes>
...other applet-specific lines...
<IMG src="URL" height=imageHeight width=imageWidth>
</APPLET>
```

Eticheta `<APPLET>` este oricare dintre etichetele container de conținut special, iar etichetele `<PARAM>` sunt oricare dintre etichetele subordonate containerului. Din cauza stilului și a convenției, eticheta `<IMG>` este plasată imediat înainte de eticheta container de închidere, iar mărimea conținutului și cea a imaginii au aceeași valoare.

Când acest set de etichete este întâlnit de un browser care nu are suport pentru miniaplicații, iată ce se întâmplă:

1. Eticheta container de deschidere nu este înțeleasă, așa că browserul o ignoră și nu face nimic.

2. La fel, etichetele următoare, care acceptă eticheta container, sunt ignorate. Până acum, browserul nu a făcut nimic altceva decât să arunce materialul pe care nu l-a înțeles.

3. Când este întâlnită eticheta `<IMG>`, browserul are brusc ceva cu care poate lucra. Interpretează această etichetă și încarcă imaginea în spațiul din document care era conceput pentru conținutul specializat.

4. Este întâlnită eticheta container de închidere. Deoarece aceasta închide ceva ce browserul nu a înțeles, este de asemenea ignorată.

Această tehnică asigură ceva de văzut tuturor celor care vă vizitează pagina, chiar dacă este un desen mare care spune „You need a Java-compatible browser for this page" (Aveți nevoie de un browser compatibil Java pentru această pagină). Altă metodă obișnuită este să folosiți o captură de ecran dintr-un film sau un fișier VRML (Virtual Reality Modeling Language) pentru imagine. Deși imaginea este statică, utilizatorul își poate face totuși o idee despre ceea ce ați vrut să prezentați. Mai multe informații despre această tehnică puteți găsi în Capitolul 16.

Rezumat

Folosirea imaginilor inline în paginile Web este unul dintre motivele pentru care World Wide Web a devenit un loc atât de căutat pentru petrecerea timpului. Pentru a adăuga aceste imagini, aveți eticheta `<IMG>`; folosirea sa depinde de un singur atribut, `src`, care identifică fișierele grafice. Deși plug-in-urile și diferite browsere acceptă o varietate mare de formate de imagine, GIF și JPEG sunt cele mai folosite și sunt acceptate în aproape orice configurație.

În HTML 4.0, au fost extinse oficial capacitățile etichetei `<IMG>` pentru a face mai ușoară controlarea înfățișării și a comportamentului graficii pe o pagină Web, la un loc cu reglajele fine pe care le puteți realiza prin intermediul foilor de stil. În același timp, a fost anunțat începutul sfârșitului pentru `<IMG>`, o dată cu suportul pentru inserarea imaginilor cu eticheta `<OBJECT>`.

Atributele pentru mapare setează desenele ca pe o hartă imagine. Hărțile imagine grafice sunt una dintre cele mai folosite întrebuințări ale fișierelor grafice, cum ar fi barele de meniu și imaginile pe tot ecranul pentru navigarea prin sit.

PARTEA a III-a

Interactivitatea de bază a HTML-ului 4.0

Capitolul 12
Să înțelegem locatorii uniformi de resurse

de David și Rhonda Crowder

În acest capitol

- **Adrese URL absolute 156**
- **Adrese URL relative 157**
- **Adrese URL cu fragment 160**
- **Tipuri de scheme pentru adrese URL 161**

Un *locator uniform de resurse* (Uniform Resource Locator), de obocei abreviat URL, specifică locația exactă a unei resurse (în mod obișnuit un fișier) pe Internet. Adresele URL sunt un subset al unui plan mai mare, bazat pe identificatori universali de resurse (Universal Resource Identifiers – URI. Până acum, adresele URL sunt singurul proiect deja instituit care face parte din conceptul URI, așa că termenii sunt uneori folosiți unul în locul altuia.

OBSERVAȚIE

În planul original, adresele URI au fost rezervate pentru schemele de localizare viitoare, iar adresele URL au fost destinate schemelor deja utilizate pentru accesarea resurselor Internet. Pentru a amesteca și mai tare terminologia adresării pe Internet, în loc de Universal Resource Identifiers se folosește uneori și „Uniform Resource Identifiers", iar în loc de Uniform Resource Locators se utilizează „Universal Resource Locators", chiar și în documentele oficiale. În afară de cei doi termeni deja cunoscuți, URI și URL, mai sunt folosiți ocazional și *Universal Document Identifiers* (UDI – identificatori universali de document) și *Uniform Resource Names* (URN – nume universale de resurse). Ultimul acronim se referă din punct de vedere tehnic numai la adresele URL care sunt obiecte persistente.

Sunt trei tipuri diferite de adrese URL: absolute, relative și cu fragment.

Adrese URL absolute

Adresele URL absolute folosesc o adresă completă de Internet pentru a furniza localizarea unei resurse. Pentru a face acest lucru, specifică întâi protocolul de transfer, apoi serverul pe care se află fișierul și, în cele din urmă, calea reală către fișierul de pe server. Adresa URL următoare, de exemplu, va fi interpretată de un browser Web ca o instrucțiune de a folosi Protocolul de Transfer Hipertext (HTTP) pentru a se duce la serverul Web Macmillan

Computer Publishing (`www.mcp.com`) și de a reda pagina de bază din directorul Sams Publishing Company (`/sams/home.html`):

```
http://www.mcp.com/sams/home.html
```

În termenii specificației oficiale (RFC 1738), referirea la protocolul de transfer se face ca fiind „schema" prin care este transferată informația. Referirea la server se face ca fiind „gazda pe rețea" a resurselor și poate fi accesat prin adrese de tip nume de domeniu sau IP (Protocol Internet), numerice. Așadar, adresa Web folosită în exemplul anterior ar putea fi scrisă în felul următor, fiind în continuare o adresă URL validă:

```
http://206.246.150.10/sams/home.html
```

OBSERVAȚIE

O dată cu apariția serverelor virtuale, gazda nu mai trebuie să fie un calculator separat. Din punctul de vedere al adreselor URL, o adresă de tip nume de domeniu sau IP care reprezintă unul dintre multele servere virtuale de pe un singur server real nu este diferită de o adresă de Internet care este unică pentru o anumită mașină.

Adrese URL relative

Adresele URL relative (uneori numite „Adrese URL parțiale") sunt folosite pentru accesarea fișierelor când nu este necesară adresa completă de Internet. Acesta este cazul majorității siturilor Web, unde cele mai multe hiperlegături sunt orientate către documente și alte fișiere aflate pe același sit și, foarte des, chiar în același director de pe situl respectiv. Se numesc adrese URL relative deoarece sunt luate în considerare față de adresa URL a documentului în care sunt înglobate (numit „document de bază"). Când sunt accesate adresele URL relative, partea de bază a adresei URL este automat concatenată cu adresa URL relativă de către browser (sau alt agent al clientului), astfel încât nu trebuie specificate separat. Adresele URL relative pot fi folosite în orice situație în care pot fi folosite adresele URL absolute (etichete ancoră, surse de imagini și altele).

Să spunem că ați ajuns la pagina de bază a sitului Web cu adresa URL absolută `http://something.com/index.html`. Deși această pagină ar putea fi legată de o alta de pe același sit cu adresa URL absolută `http://something.com/nextpage.html`, o metodă mai economică este simpla legare cu adresa URL relativă `nextpage.html`.

Broweserul taie tot ce urmează ultimului / din adresa documentului curent, apoi adaugă adresa URL relativă la sfârșitul adresei URL de bază care a rezultat. În acest caz, ar înlătura partea cu `index.html` din adresa URL absolută `http://something.co/index.html` pentru a obține adresa URL de bază `http://something.com/` și ar înlocui porțiunea înlăturată cu adresa URL relativă `nextpage.html`. Rezultatul este adresa URL completă `http://something.com/nextpage.html`. Browserul poate găsi acum pagina corectă ca și cum în pagina Web apelantă ar fi fost codată adresa URL absolută a paginii următoare.

SUGESTIE

Adresele URL relative sunt o necesitate absolută dacă doriți ca situl dumneavoastră Web să fie cu adevărat portabil. De exemplu, un sit cu instrucțiuni care este comprimat și disponibil pentru descărcare și instalare pe propriul sistem al vizitatorului nu ar avea nici un folos dacă ar folosi adrese URL absolute pentru fiecare legătură internă a sa, deoarece acestea ar cere accesul la serverul original ori de câte ori este urmată o legătură. Veți dori, de asemenea, să folosiți adresele URL relative în orice situație asemănătoare. Siturile-oglindă, de exemplu, au nevoie să facă copii ale fișierelor disponibile fără a cauza accesări ale serverului original. De asemenea, dacă vă mutați pagina de bază pe un nou server, este mult mai ușor să vă schimbați adresa de Internet dacă nu este nevoie să recodați fiecare legătură internă.

(continuare)

(continuare)
Dacă, pe de altă parte, nu doriți ca situl dumneavoastră să fie portabil, ar trebui să evitați adresele URL relative și să lucrați doar cu adrese URL absolute.

Adresele URL relative pot indica fișiere din același director cu documentul apelant, dintr-un subdirector sau dintr-un director tată. Nu puteți, bineînțeles, folosi adrese URL relative pentru o resursă care nu are aceeași adresă URL de bază cu documentul care o invocă.

Pentru a coda o adresă URL relativă către un fișier în același dirctor cu documentul care îl invocă folosiți numele acelui fișier, după cum am arătat la începutul acestei secțiuni.

Pentru o adresă URL relativă către un fișier (în acest caz `lowerdoc.html`) dintr-un subdirector al celui care conține documentul apelant, procedura este aceeași, cu excepția că trebuie să adăugați și numele subdirectorului (în acest caz `lowerdir`) și, bineînțeles, un slash de separație între subdirector și numele fișierului, ca în exemplul următor:

```
lowerdir/lowerdoc.html
```

Pentru a folosi o adresă URL relativă către un fișier (în acest caz `uperdoc.html`) din directorul tată al celui care conține documentul apelant, procedați la fel ca pentru un subdirector, dar folosiți componenta de cale `..` pentru numele directorului, ca în exemplul următor:

```
../upperdoc.html
```

Deși aceste exemple au folosit drept resurse fișiere HTML, toate resursele adresabile sunt identificate în același fel. Un fișier imagine numit `face.gif`, de exemplu, dintr-un subdirector numit `images`, ar fi indicat cu adresa URL relativă `images/face.gif`.

Stabilirea adresei URL de bază

Adresa URL de bază, față de care sunt calculate adresele URL relative, nu este de obicei specificată de autorul codului HTML. În mod prestabilit, ea este pur și simplu locația documentului apelant sau ia valoarea din entitatea `Content-Base` din antetul HTTP (vezi secțiunea „Protocolul de transfer al hipertextului (HTTP)").

Totuși, autorii de cod HTML pot și chiar schimbă adresa URL de bază, implicit și explicit. Dacă folosiți „`Refresh`" drept valoare a atributului `http-equiv` al elementului `META` pentru a încărca o pagină nouă, atunci adresa URL specificată în acel element stabilește noua adresă URL de bază. De exemplu, să spunem că aveți o pagină care a fost mutată de la `http://www.oldsite.com/index.html` și ați aranjat o redirecționare de la situl vechi la cel nou, `http://www.newsite.com/index.html`. În acest caz, pe situl vechi ar fi trebuit să aveți un cod asemănător cu acesta:

```
<HEAD>
<META http-equiv="Refresh" content="3; url=http://www.newsite.com/index.html">
</HEAD>
```

Acest cod va redirecționa browserul dumneavoastră către noul sit la trei secunde după ce a fost încărcat cel vechi, iar adresa URL de bază va fi, desigur, modificată și ea.

ATENȚIE!

Specificația W3C pentru HTML 4.0 greșește declarând că precedentul cod ar trebui să fie sub forma următoare:

```
<META name="refresh" content="3,http://www.newsite.com/index.html">
```

Am încercat această variantă și nu merge nici pe Netscape Navigator, nici pe Internet Explorer. Folosiți varianta cu `http-equiv`.

Mai sunt câteva metode prin care puteți declara explicit adresa URL de bază pentru un document. Atributul `http-equiv` al elementului `META` poate fi folosit pentru a anula entitățile existente în antetul HTTP, astfel încât să puteți realiza un cod asemănător cu următorul:

```
<HEAD>
<META http-equiv="Content-Base" content="http://www.newsite.com/">
</HEAD>
```

Aceasta însă sigur *nu* este o abordare bună. Deși adăugarea propriilor dumneavoastră anteturi personalizate nu este o problemă și va fi acceptată de majoritatea browserelor, orice încercare de a anula anteturile în timpul interacțiunii normale dintre agentul clientului și serverul HTTP (chiar dacă serverul o va permite, dar e posibil să nu o facă) va rezulta probabil într-un comportament imprevizibil.

O metodă mult mai bună de a stabili adresa URL de bază este folosirea elementului `BASE`, care este creat special pentru a realiza această funcție. Fiecare document poate avea propriul său element `BASE`, deoarece domeniul său de existență este limitat la documentul care îl conține. Elementul `BASE` este conținut în elementul `HEAD` și este folosit după cum urmează:

```
<HEAD>
<BASE href="http://www.thissite.com/">
</HEAD>
```

Observați că `BASE` nu are etichetă de închidere; nu există `</BASE>`.

OBSERVAȚIE

Deși este dificil să ne imaginăm de ce ar avea cineva mai mult de un element `BASE` în `HEAD`, specificația W3C afirmă că ultimul element `BASE` stabilește noua adresă URL de bază. Trebuie să menționăm totuși că un `META http-equiv="Refresh"` – înainte sau după orice element `BASE` – contrazice această afirmație, deoarece, redirecționând browserul, se anulează elementul `BASE`.

În plus față de `href`, `BASE` are și atributul `target`, care arată cadrul în care documentul invocat va fi afișat. Acest atribut nu are nevoie să fie specificat decât dacă folosiți mai multe cadre și doriți să încărcați resursele apelate de adresele URL relative într-unul anume dintre acestea. Dacă (după cum este cazul de obicei) acesta nu este specificat, atunci `target` ia valoarea prestabilită `_self`, ceea ce înseamnă că un document legat va ocupa cadrul din care a fost apelat (dacă nu ați folosit cadre, documentul apelat o să ocupe întreaga fereastră a browserului).

Dacă atributul `target` are avloarea `_blank`, indiferent dacă folosiți cadre sau nu, rezultatul este deschiderea unei noi ferestre a browserului în care va fi încărcat documentul apelat. Codul HTML necesar pentru a face acest lucru este următorul:

```
<HEAD>
<BASE target="_blank">
</HEAD>
```

Mai sunt încă două nume rezervate pe care le puteți invoca pe post de `target`. Acestea sunt `_parent` și `_top`. `_parent` și fac astfel încât documentul apelat să fie afișat în `FRAMESET`-ul care conține cadrul apelant, iar `_top` face ca documentul să fie afișat în toată fereastra browserului, eliminând astfel toate cadrele.

De asemenea, puteți folosi ca valoare a atributului `target` orice nume personalizate pe care le-ați atribuit cadrelor dumneavoastră. Codul HTML de mai sus, pentru valoarea `_blank`, poate fi folosit cu oricare dintre aceste nume, fie el rezervat sau personalizat.

Cele două metode pot fi combinate, specificând adresa URL de bază și cadrul țintă. Puteți chiar să folosiți o altă adresă URL de bază decât cea în care este rezident documentul apelant împreună cu o pereche de cadre cu nume personalizate și să configurați într-un cadru un meniu de legături care folosește adrese URL relative, pentru a apela numai pagini din alt sit și a le afișa în al doilea cadru.

În acest caz elementul BASE va arăta asemănător cu cel din exemplul următor:

```
<HEAD>
<BASE href="http://www.anothersite.com/" target="display_window">
</HEAD>
```

Orice țintă specificată în elementul BASE poate fi anulată în funcție de caz prin specificarea altei ținte într-o ancoră, legătură, hartă imagine sau formular.

ATENȚIE!

Dacă declarați explicit adresa URL de bază în secțiunea HEAD, veți face ca situl să nu fie portabil, obținând același efect ca în cazul în care ați fi codat fiecare adresă URL relativă ca pe una absolută. Dacă vreți ca situl să funcționeze pe alt server sau pe alt sistem local, ar trebui să înlăturați din secțiunea HEAD informațiile explicite privind calea de bază în versiunea pe care vreți s-o faceți disponibilă pentru descărcare sau să le adaptați dacă vă mutați situl la altă adresă de Internet.

Adrese URL cu fragment

Folosirea atributului name al unui element <A> sau a atributului id al oricărui element creează o ancoră. Aceste ancore pot fi referite în maniera semnelor de carte (bookmark) folosind adrese URL cu fragment. De exemplu, o ancoră ar putea fi declarată într-unul din felurile următoare:

```
<A name="candybar">This is the candy bar anchor. </A>
<H1 id="candybar">This is the candy bar anchor.</H1>
```

OBSERVAȚIE

Nu puteți folosi amândouă metodele cu același identificator de fragment, deoarece atât name, cât și id ocupă același spațiu pentru nume. Aceasta înseamnă că o variabilă atribuită unuia nu poate fi atribuită și celuilalt.

Adresa de URL cu fragment pentru a accesa o astfel de ancoră poate arăta în felul următor:

```
http://www.fragdemo.org/chocolate.hthml#candybar
```

Porțiunea care urmează adresei paginii Web și începe cu diez (#) se numește „fragment", de unde și numele de *adresă URL cu fragment*. Partea fragmentului de după diez (în acest caz, candybar) se numește *identificator de fragment* și are aceeași valoare ca și atributul name sau id din ancora în discuție.

Nici # și nici identificatorul de fragment care îi urmează nu fac, din punct de vedere tehnic, parte din URL deoarece acestea nu au fost niciodată incluse în schema oficială HTTP. De fapt, adresele URL cu fragment, deoarece au această porțiune nonstandard la sfârșitul lor, nu sunt, oficial, adrese URL, deși așa sunt tratate de browserele Web și pot fi considerate ca atare pentru toate scopurile de către autorii de cod HTML.

Adresele URL cu fragment pot fi absolute, ca în exemplul anterior, sau relative, ca în acest exemplu:

```
chocolate.html#candybar
```


Domeniul de existență al identificatorului de fragment este limitat la cadrul documentului în care se află. Acest lucru înseamnă că există un tip neoficial de adresă URL „relativă în cadrul documentului", care poate fi realizată numai cu aceste ancore. Similar cu o adresă URL relativă, care nu are nevoie de o adresă URL de bază declarată explicit, nici adresa URL cu fragment nu are nevoie de o prefață *dacă legătura către ea este în documentul în care se află*. Acesată adresă URL „relativă la document" arată cam așa:

```
#candybar
```

Tipuri de scheme pentru adrese URL

Deși există mai multe protocoale de transfer și scheme diferite, de la cele vechi, ca Gopher și WAIS, la cele obscure, ca Prospero, autorii de cod HTML sunt de obicei preocupați numai de cele câteva scheme de adrese URL folosite de obicei. Cele mai des folosite sunt HTTP (pentru pagini Web), mailto (pentru e-mail), news (pentru grupuri de informare) și, mai puțin folosită în zilele noastre deoarece fișierele pot fi transmise prin HTTP, FTP (pentru transferul fișierelor). Pentru ca lucrarea aceasta să fie cât mai completă, vom furniza câteva informații și despre cele care sunt mai puțin cunoscute, fie din cauza utilității limitate, fie datorită vechimii, cum ar fi file și telnet. Vom menționa, de asemenea, altele, cum ar fi protocolul TV, create pentru a folosi noile tehnologii care au apărut sau vor apărea în curând.

Majoritatea schemelor URL au elemente opționale, pe care nu le veți folosi de obicei. Porturile, de exemplu, nu sunt specificate de obicei deoarece majoritatea administratorilor de rețea le acceptă pe cele standard. Așadar, un server Web este aproape întotdeauna pe portul 80 (deși ocazional puteți găsi unul pe un alt port, cum ar fi 8001).

Protocolul de transfer al hipertextului (HTTP)

Schema de adrese URL care prezintă cel mai mare interes pentru autorii de cod HTML este, desigur, HTTP (Hypertext Transfer Protocol), care este protocolul nativ al Webului. Acesta este folosit pentru a accesa pagini HTML prin intermediul browserelor și al altor agenți ai clienților. HTTP folosește adresele URL în câteva moduri foarte interesante.

Schema de adresă URL pentru HTTP arată în felul următor:

```
http://<host>:<port>/<path>?<searchpart>
```

Prima parte (http://) definește protocolul folosit pentru accesarea gazdei (host). *host* este fie un nume de domeniu, cum ar fi www.mcp.com, fie o adresă numerică IP, ca 206.246.150.10. După cum am mai spus, valoarea prestabilită pentru port este 80 în cazul serverelor Web. *path* definește localizarea documentului HTML căutat pe gazdă. *searchpart*, care este opțional, include parametrii pentru procesarea CGI.

În procesarea HTTP, agentul clientului (de obicei un browser Web, cum este Netscape Navigator) trimite o cerere serverului HTTP. Prima componentă a cererii este *metoda*, care este urmată de alte informații despre antet. Cele mai folosite metode sunt GET, HEAD și POST. GET este probabil cea mai folosită metodă, deoarece ea reprezintă felul în care se cere o pagină Web. Un antet simplu de cerere GET arată cam în felul următor:

```
GET /sams/home.html HTTP/1.1
Connection: Keep-Alive
User-Agent: Mozilla/3.01Gold(Win95;I)
Host: www.mcp.com
Accept: image/gif, image/jpeg
```

Prima parte cere serverului HTTP să trimită o pagină browserului. De asemenea furnizează versiunea HTTP. Cerând serverului să păstreze linia deschisă, îi comunică ce browser (`User-Agent`) și ISP sunt folosite și ce alte tipuri de fișiere acceptă.

Porțiunea `sams/home.html` se numește `Request-URI`. În acest caz, este o adresă URL relativă, compusă din calea de la URL, care este calea locală completă de pe server. Aceasta este cea mai folosită formă a unei cereri `GET` către un server. O cerere `GET` care trebuie să treacă printr-un server proxy trebuie să folosească adresa URL absolută a resursei, care o să includă informațiile gazdei și o să fie prefațată de `http://`.

Serverul răspunde prin trimiterea unui antet cu informații de identificare similare la un loc cu fișierul cerut (care se numește *corp entitate*) sau, dacă se întâmplă ceva în neregulă, un mesaj de eroare. De exemplu, dacă pagina solicitată nu există, vă va apărea pe ecran răspunsul `404: Not Found`.

Metoda `HEAD` prezintă un interes mai scăzut din partea autorilor de cod HTML. Funcționează în același fel ca și metoda `GET`, dar nu este returnat nici un fișier din partea serverului, doar informații despre fișierul specificat. Această metodă este folosită de agenții clienților pentru a determina, de exemplu, ultima dată la care a fost revizuită pagina HTML, astfel încât să poată decide folosirea versiunii din cache sau, dacă aceasta este mai veche, crearea uneia mai noi de la server.

Metoda `GET` mai poate fi folosită pentru a activa un program CGI la server prin folosirea `searchpart`-ului din URL. De obicei, `searchpart` este folosit cu intrarea formularelor (vezi Capitolul 15, „Crearea și utilizarea formularelor HTML"). În acest caz, primul rând al antetului va arăta în felul următor:

```
GET /cgi-bin/cats.pl?breed=shorthair&age=4 HTTP/1.1
```

În loc să returneze o pagină HTML deja existentă, această metodă invocă un program Perl CGI numit `cats.pl` în directorul `cgi-bin` al serverului. Acel program va procesa informațiile pe criterii de fizionomie (breed) și vârstă (age). Apoi, ar crea o nouă pagină HTML ca răspuns la cerere și ar trimite-o înapoi serverului, care ar retransmite pagina browserului de pe care a fost formulată cererea.

Metoda `POST` execută o sarcină similară, dar în alt mod. Metoda `GET` trimite variabilele introduse ca parte a adresei URL propriu-zise, dar metoda `POST` lucrează puțin diferit. După cum sigur vă amintiți, o cerere conform metodei `GET` este formată din linia metodei urmată de alte informații. O cerere care folosește metoda `POST` adaugă o a treia parte, coprul entitate. Ați întâlnit deja corpul entitate în forma paginii HTML returnate în urma unei cereri `GET`. Un corp entitate `POST` este același lucru, dar mergând în altă direcție. În locul unei pagini HTML, corpul entitate `POST` constă în parametrii trimiși programului CGI, iar adresa URL dintr-o cerere `POST` nu conține un `searchpart`.

OBSERVAȚIE

Din cele trei metode folosite de obicei, numai `GET` și `HEAD` trebuie, de fapt, să fie acceptate de toate serverele HTTP. Este posibil, deși puțin probabil, ca un server HTTP să nu accepte metoda `POST`.

Mailto

Folosiți de obicei adresele URL `mailto` pentru a invoca programele e-mail (cum e banala legătură `Contact the Webmaster`, găsită în josul aproape fiecărei pagini Web). Schema de adresă URL este după cum urmează:

```
mailto:<e-mail address>?<subject=>
```


Observați că nu există slash dublu după caracterul „două puncte", așa cum este cazul în adresele URL HTTP.

Când este apelată, ea lansează un program de e-mail cu câmpul `To` umplut cu specificația de *`e-mail address`*. La fel ca la `searchpart` din adresele URL HTTP, puteți specifica o adăugare opțională – în acest caz, `subject`. Dacă este specificat un subiect (subject) în URL, atunci câmpul `Subject` din e-mail este umplut cu valoarea respectivă.

Deci, o adresă URL mailto ca `mailto:john@smith.com?subject=Happy Birthday` va crea un mesaj e-mail pregătit să transmită felicitări de ziua de naștere persoanei în cauză la adresa respectivă.

SUGESTIE

Internet Explorer nu poate manevra adresele URL `mailto` și chiar s-ar putea bloca dacă le întâlnește.

Într-o adresă URL `mailto` nu este specificat nici un port. În acest moment se poartă unele discuții despre extinderea adreselor URL `mailto` pentru a include și alte opțiuni, cum ar fi `<body=>`, care se vor comporta la fel ca `<subject=>`, dar vor plasa automat mesaj e-mail propriu-zis. Această caracteristică ar fi foarte folositoare pentru trimiterea listelor, deoarece ar permite foarte ușor (și fără greșeli) comandarea subscrierilor și anulărilor de pe o pagină Web.

News

Poate doriți să folosiți și adresele URL `news`, pentru ca vizitatorii sitului dumneavoastră să poată accesa grupuri de informare Usenet legate. Deși furnizați adresa URL pe situl dumneavoastră, șansa vizitatorilor de a stabili contactul cu grupurile de informare depinde de propriul lor acces la un server de informare (de obicei furnizat de ISP-ul lor). O adresă URL `news` pur și simplu furnizează cititorului de informații parametrii pentru a localiza fie un anume grup de informare, fie un anume articol, dacă aveți mesajul ID pentru el. Schema adreselor URL `news` are două versiuni:

```
news:<newsgroup-name>
news:<message-id>
```

Prima versiune este cea folosită în mod obișnuit. Pentru a vă conecta la grupul de informare Archaeology, de exemplu, puteți coda următoarele într-o legătură:

```
news:sci.archaeology
```

A doua versiune specifică articole individuale în locul întregului grup de informare. Însă această versiune are o utilitate limitată, deoarece durata de viață a unui articol este efemeră și obținerea ID-ului mesajului său implică un efort suplimentar.

La fel ca la adresele URL `mailto`, nu este implicat nici un port și nici nu sunt două slashuri după caracterul „două puncte".

Protocolul de transfer al știrilor în rețea (NNTP)

Ca o variantă la specificarea unui grup de informare Usenet, puteți folosi Network News Transfer Protocol (NNTP – protocolul de transfer al informațiilor în rețea) pentru accesarea grupurilor de informare. Iată schema de adresă URL pentru NNTP:

```
nntp://<host>:<port>/<newsgroup-name>/<article number>
```

Dar cum majoritatea serverelor de informații nu permit accesul oricui, această adresă URL are o utilitate foarte limitată. Valoarea prestabilită a portului este 119. Numele grupului de

informare este tratat la fel ca o adresă URL `news`, iar numărul articolului este mesajul ID. De asemenea, dacă articolul a expirat, legătura nu mai poate fi folosită.

Protocolul de transfer al fișierelor (FTP)

File Transfer Protocol (Protocolul de transfer al fișierelor) este încă folosit pentru recuperarea fișierelor de pe serverele FTP, deși nu în aceeași măsură ca în zilele trecute, datorită capacității HTTP-ului de a transfera fișiere. Aceasta este schema de adresă URL pentru specificarea unei resurse de pe un server FTP:

```
ftp://<user>:<password>@<host>:<port>/<cwd1>/ <cwd2>/.../<cwdN>/
<name>;type=<typecode>
```

Un autor de cod HTML, desigur, nu furnizează numele sau parola unui utilizator când codează o legătură către o adresă URL FTP. În cele mai multe cazuri, acestea nu sunt necesare, oricum, deoarece majoritatea siturilor FTP către care adresați legături ar fi situri FTP anonime, care sunt deschise pentru toată lumea. Valoarea prestabilită a portului este 21. Părțile *cwd* sunt calea directoarelor către fișier, iar *name* este numele fișierului căutat.

Vorbind în general, tot ce aveți de făcut dumneavoastră, ca autor de cod HTML, este să furnizați protocolul fișierului, gazda și adresa fișierului, iar programul are grijă de rest. Așadar, puteți simplifica sintaxa precedentă în felul următor:

```
ftp://<host>/<cwd1>/<cwd2>/.../<cwdN>/<name>
```

Nu trebuie să specificați un anume fișier. Puteți doar furniza gazda, după cum se vede în exemplul următor, ceea ce va cauza o listare a directoarelor de pe situl căutat:

```
ftp://ftp.microsoft.com
```

File

Adresele URL `file` sunt rar întâlnite cu excepția cazului în care accesați o pagină Web pe care ați descărcat-o pe o unitate locală. De fapt, adresele URL `file` nu au nici un protocol stabilit oficial asociat lor, dar totuși ele funcționează ca un înlocuitor viabil pentru FTP și pot fi folosite la recuperarea fișierelor de pe servere FTP anonime.

Iată schema adresei URL `file`:

```
file://<host>/<path>
```

Un exemplu de folosire a unei scheme de adresă URL `file` ca FTP ar arăta cam așa:

```
file://ftp.microsoft.com/
```

Adresele URL `file` nu sunt cu adevărat necesare în creațiile HTML, deși au un rol tangențial dacă intenționați să folosiți tehnologiile Web în afara rețelei World Wide Web. O pagină Web aflată pe un CD-ROM, de exemplu, are o adresă URL `file`. Însă nu aveți nevoie să codați această informație pe nici una dintre paginile dumneavoastră, deoarece pagina ar putea fi accesată prin metodele de deschidere a fișierelor ale unui browser Web. În acest caz, adresa URL `file` pur și simplu apare în caseta Location din partea superioară a browserului.

Telnet

Telnet este un mijloc de a vă conecta la un sistem aflat la distanță. Schema adresei URL `telnet` este după cum urmează:

```
telnet://<user>:<password>@<host>:<port>/
```

La fel ca în cazul FTP-ului, nu trebuie să specificați numele sau parola utilizatorului într-o legătură. Valoarea prestabilită a portului este 23. Iată un exemplu despre cum să folosiți o schemă de adresă URL `telnet` pentru a găsi situl Bibliotecii Congresului:

```
telnet://locis.loc.gov
```


Caractere rezervate sau nesigure

Sunt mai multe caractere care nu ar trebui folosite într-o adresă URL, fie pentru că pot cauza conflicte în interpretarea adreselor URL, fie că sunt rezervate de diferite scheme de adrese URL. De exemplu, diezul (`#`), deoarece este folosit pentru a specifica o adresă URL cu fragment, nu ar trebui niciodată folosit în alt fel. Adresa URL `http://www.unsafe.com/index#1.html` ar putea cauza probleme dacă ar fi interpretată ca o adresă URL cu fragment. Nu există nici un motiv pentru care un caracter să nu fie folosit în locul potrivit. Totuși, dacă un caracter va fi folosit într-o adresă URL într-un mod diferit de cel prevăzut de schema de adrese URL, trebuie să fie specificat cu un *cod escape* folosind notația `%HH` (hexazecimal). Următorul tabel ilustrează aceste caractere și echivalențele lor codate.

Tabelul 12.1. Coduri de evadare pentru caractere nesigure sau rezervate.

Caracter	*Codul escape*
Space	%20
"	%22
#	%23
%	%25
&	%26
/	%2F
:	%3A
;	%3B
<	%3C
=	%3D
>	%3E
?	%3F
@	%40
[	%5B
\	%5C
]	%5D
^	%5E
`	%60
{	%7B
|	%7C
}	%7D
~	%7E

Tendințe viitoare

În continuare sunt propuse noi scheme de adrese URL. Multe sunt neînsemnate sau nu prezintă interes pentru autorii de cod HTML, dar sunt două scheme interesante care sigur vor intra pe scenă în curând. Una este schema de adrese URL `TV`, pentru recepția programelor de televiziune. Formatul său este următorul:

`tv:<broadcast>`

Iată câteva exemple de folosiri propuse pentru ea:

- **`tv:cnn`** Pentru folosirea cu ID-uri netword
- **`tv:wsvn`** Pentru folosirea cu semnale de apel
- **`tv:10`** Pentru folosirea cu numerele canalelor

Cealaltă viitoare schemă de adrese URL, similară cu schema `mailto`, este schema `callto` pentru comunicațiile telefonice. Are câteva variante propuse, inclusiv pentru voce, fax sau comunicarea datelor, pentru pauze adaptabile sau pentru adăugarea extensiilor de formare după conectare. Formatul de bază pentru adresa URL `callto` este următorul:

`callto:<phone number>`

Rezumat

Locatorii uniformi de resurse, cunoscuți printr-o varietate de denumiri similare, sunt tehnici pentru stabilirea adreselor Internet pentru fișiere și celelalte resurse. Există trei tipuri de adrese URL: absolute, relative și cu fragmente. Adresele URL absolute conțin adrese complete, inclusiv protocolul, gazda, calea și numele fișierului. Adresele URL relative nu au nevoie să fie specificată gazda sau, în unele cazuri, calea. Adresele URL cu fragment fac referire la elemente numite într-o pagină HTML particulară. Sunt mai multe tipuri de formate ale adreselor URL, numite „scheme". Printre ele se numără `HTTP`, `mailto`, `news` și `FTP`. Mai multe caractere sunt rezervate de schemele de adrese URL sau ar putea intra în conflict cu interpretarea corectă a adreselor URL, așa încât ele trebuie specificate prin coduri de evadare. Vi s-au prezentat, de asemenea, unele scheme de adrese URL propuse care i-ar putea afecta pe autorii de cod HTML. Înțelegerea completă a diferitelor adrese URL și a folosirii lor corecte este esențială pentru un design bun pentru Web. În capitolul următor, „Hiperlegături și ancore", vom arunca o privire la aplicațiile practice ale adreselor URL.

Capitolul 13
Hiperlegături și ancore

de David și Rhonda Crowder

În acest capitol

- **Întrebuințările hiperlegăturilor 167**
- **Structura hiperlegăturilor 168**
- **Legături către un conținut specializat 173**

Hipertextul reprezintă esența și inima World Wide Web-ului. Fără hiperlegături (legături la hipertext), Webul nu ar fi nimic altceva decât niște fișiere separate și fără nici o legătură unul cu celălalt, față de totul coerent și interconectat pe care îl reprezintă. Legarea diferitelor fișiere într-o singură unitate poate fi văzută la scară mică în compunerea unei pagini HTML, pe o scară mai mare în construcția unui sit Web, iar la scară macro în totalitatea Webului însuși.

ATRIBUTE COMUNE

Majoritatea elementelor din acest capitol au atribute HTML comune, cum ar fi articolele care verifică identificarea, stilul, limbajul și tratarea evenimentelor. Deoarece atributele comune sunt atât de des întâlnite în HTML 4 (de aceea sunt numite „comune"), veți găsi o prezentare a lor în Introducere, la începutul acestei cărți. Puteți de asemenea găsi mai multe informații despre aceste caracteristici des întâlnite și elementele lor asociate în Anexa A.

Între timp, în acest capitol, mă voi referi la aceste atribute comune folosind următoarele cuvinte cheie: core (principale), international (internaționale) și events (evenimente).

În plus, multe elemente pot accepta alte elemente în cadrul granițelor lor. Acestora li se va spune *elemente inline* și includ următoarele opțiuni:

Elemente `font`: Aceste elemente includ etichetele pentru bold (`<B>`) și italic (`<I>`) și sunt discutate în Capitolul 17, „Stil simplu cu text".

Elemente `phrase`: Aceste elemente includ etichete cum ar fi cele de evidențiere (`<EM>`), cod (`<CODE>`) și adresă (`<ADDRESS>`) și sunt discutate în Capitolul 7, „Alinierea și formatarea textului".

Elemente `special`: Această colecție este o amestecătură de elemente care include următoarele: miniaplicații și obiecte (Capitolul 16, „Obiecte înglobate în HTML"), imagini (Capitolul 11, „Adăugarea imaginilor pe pagina dumneavoastră Web"), ancore (Capitolul 13, „Hiperlegături și ancore"), hărți imagine (Capitolul 14, „Crearea meniurilor cu hărți imagine"), scripturi (Capitolul 23, „Scurtă privire asupra limbajului JavaScript" și 24, „Folosirea limbajului VBScript"), fonturi, superscripturi și subscripturi (Capitolul 17), salturi la rând nou și citate scurte (Capitolul 7) și cadre inline (Capitolul 20, „Crearea cadrelor").

Elementele `form`: Acest set este folosit pentru a adăuga elemente unui formular, inclusiv butoane, câmpuri de text și casete de validare.

Întrebuințările hiperlegăturilor

Hiperlegăturile sunt folosite în mai multe feluri diferite:

- Conectarea cu alte pagini HTML care fac parte din situl Web sau cu materiale din alte situri Web. Această întrebuințare este cel mai des implementată prin folosirea

elementului ANCHOR, deși elementul LINK s-ar putea dovedi, în cele din urmă, mai util (vezi Capitolul 7 pentru informații detaliate despre elementul LINK).

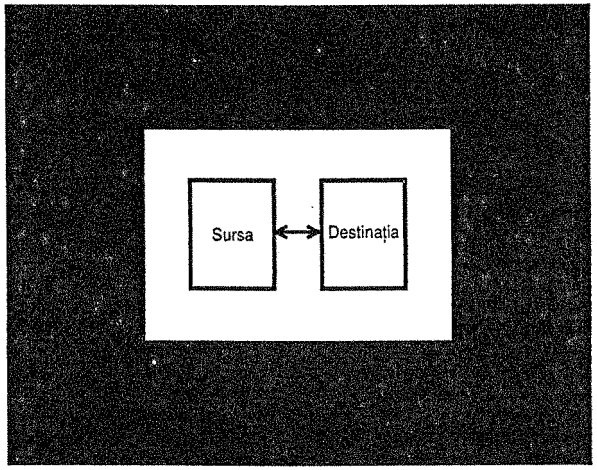
- Înglobarea obiectelor, cum ar fi imaginile și sunetele, într-o pagină HTML (vezi Capitolul 16).
- Conectarea la resurse non-HTML.
- Oferirea informațiilor adiționale; sunt folosite pentru elemente cum ar fi Q, BLOCKQUOTE, INS și DEL (vezi Capitolul 7) în vederea furnizării unei legături către alte documente care conțin mai multe informații despre întrebuințările elementului în cauză. Caracteristica informațională este nouă și slab implementată.

Hiperlegăturile sunt des folosite pentru ca un sit Web să poată fi împărțit în mai multe pagini HTML diferite, interlegate între ele. Varianta de a avea o singură pagină foarte lungă îngreunează manevrarea și ar avea un timp de încărcare inacceptabil de mare.

Structura hiperlegăturilor

Fiecare hiperlegătură are două capete: ancora sursă și ancora destinație. (Vezi Figura 13.1.) De exemplu, în legătura conținută în codul HTML `<a href="http://www.linkfinder.com/">`, destinația finală a legăturii este situl Web LinkFinder. Sursa acestei legături poate fi orice document care conține acest cod. Toate hiperlegăturile sunt niște străzi cu două sensuri; pentru o legătură dintre două pagini HTML, sursa și destinația sunt disponibile și prin executarea unui clic pe butoanele Back (Înapoi) și Forward (Înainte) ale browserului dumneavoastră Web.

FIGURA 13.1.
Structura legăturii.



OBSERVAȚIE

A-ul din eticheta <A> reprezintă cuvântul anchor (ancoră). Totuși, sunt unele dispute despre corectitudinea terminologiei. Documentația oficială a W3C despre HTML 4.0 spune că o ancoră este „o zonă denumită în cadrul unui document HTML" și că eticheta <A> este folosită la definirea unei legături. Deși este posibil să atribuiți un nume unui element ANCHOR, acest lucru nu este necesar. Deoarece tot ce are o etichetă <A> este denumit de obicei „ancoră" de majoritatea autorilor de cod HTML, vom folosi această terminologie și ne vom referi, în acest capitol, la un element ANCHOR cu un atribut name ca fiind o *ancoră denumită*.

Deși majoritatea legăturilor sunt către alte pagini HTML, în afara sau înăuntrul sitului Web pe care sunt acestea rezidente, adesea ele fac trimitere către alte tipuri de fișiere, de obicei fișiere imagine (GIF, JPEG sau PNG). Cu toate că la prima privire nu se observă o relație cu două sensuri în aceste cazuri, aceasta există; vă permite să vedeți imaginea separat, dacă așa doriți, putând apoi să vă întoarceți la pagina Web pe care este înglobată.

Relația cu două sensuri există, de asemenea, în cazuri în care nu este atât de evidentă și în care utilizatorii descoperă că este imposibil să exploreze conectarea bidirecțională, cum ar fi atunci când o miniaplicație Java este legată la o pagină. Și totuși există, pentru că miniaplicația nu trebuie doar localizată, dar trebuie să știe și unde să se execute.

Elementul ANCHOR

Elementul ANCHOR, care apare numai în corpul unei pagini HTML, este de obicei redat într-un anume fel într-un browser Web. Dacă conține doar text, este de obicei colorat în albastru și este subliniat, dar nu sunt făcute alte schimbări asupra textului. Desigur, această formatare poate fi schimbată prin folosirea foilor de stil pentru a defini diferit stilul elementului, dar dacă faceți acest lucru riscați să-i zăpăciți pe vizitatorii sitului dumneavoastră, care sunt obișnuiți să găsească reprezentarea standard. Dacă ancora conține o imagine, cu sau fără text însoțitor, imaginea este de obicei scoasă în evidență cu albastru.

ATENȚIE!

Elementul ANCHOR necesită atât eticheta de deschidere (<A>), cât și cea de închidere (</A>). Dacă nu includeți eticheta de închidere, veți realiza o hiperlegătură *foarte* lungă.

Atributul href (referință hipertext) este singurul absolut necesar într-un element ANCHOR. Fără el, hiperlegătura nu există. Un element ANCHOR fără un href specificat nu reprezintă nimic. De fapt, un browser Web nici măcar nu-l recunoaște ca fiind o ancoră. Codurile ca `<A>This is not an anchor</A>` nici măcar nu se albăstresc când sunt redate de Netscape Navigator sau Internet Explorer.

O dată ce este adăugat atributul href, ancora devine funcțională. Folosirea sa corespunzătoare este următoarea:

```
<A href=URL>Anchor text or image goes here.</A>
```

Adresa URL poate fi o adresă absolută completă, o adresă URL relativă sau chiar și o adresă URL cu fragment și trebuie închisă între ghilimele. Toate exemplele următoare sunt hiperlegături corecte:

```
<A href="http://www.someaddress.com/index.html">A link to a full URL.</A>
<A href="/index.html">A link to a partial URL.</A>
<A href="#middle">A link to a fragment URL (named anchor).</A>
<A href=http://www.someaddress.com/index.html><img src=image.gif">Image
➥link - optional text goes here.</A>
```

Foarte interesant, atributul href trebuie să fie prezent pentru ca o ancoră de hiperlegătură să fie recunoscută de browserele Web, dar adresa URL nu trebuie specificată. În Netscape Navigator, codul următor este suficient pentru ca ancora să fie recunoscută:

```
<A href>An improper link.</A>
```

Internet Explorer tot nu va recunoaște ancora în aceste condiții, dar adăugarea semnului de egal (=) va realiza același efect precum codul precedent pentru Netscape Navigator:

```
<A href=>Another improper link.</A>
```

Amândouă sunt, bineînțeles, exemple de coduri HTML foarte proaste. Singurul motiv pentru care funcționează este că browserele Web sunt concepute să fie foarte înțelegătoare. Minimul necesar pentru un standard corect ar fi următorul:

```
<A href="">An empty URL.</A>
```

Legătura creată de acest cod folosește, în mod prestabilit, adresa URL de bază. Așadar, codul anterior, dacă se află într-o pagină la `http://www.mycompany.com/clocks.html`, va face referire la `http://www.mycompany.com/` și, tot în mod prestabilit, va încărca pagina de bază.

Dacă folosiți cadre, atunci poate fi folosit atributul `target` pentru a specifica în ce cadru va fi afișată resursa conectată. Dacă ați stabilit un cadru numit `main_window`, de exemplu, atunci puteți specifica o pagină HTML pentru a fi încărcată în acesta prin folosirea codului următor:

```
<A href="bagua.html" target="main_window">Feng Shui</A>
```

Aveți mai multe opțiuni de a specifica cadrul țintă. Atributul `target` funcționează la fel în elementul ANCHOR ca și în elementul BASE. Pentru o explicare detaliată a atributului `target`, vezi Capitolul 12, „Să înțelegem locatorii uniformi de resurse".

Atributele `rel` și `rev` îndeplinesc exact aceeași funcție în elementul ANCHOR pe care o îndeplineau și în elemntul LINK (vezi Capitolul 7). Atributul `accesskey` este discutat în Capitolul 41.

Ancore denumite

Ancorele denumite sunt acele elemente care conțin fie o etichetă <A> cu un atribut `name`, fie orice alt element cu un ID atribuit. Se poate folosi sau atributul `name`, sau `id`, deoarece nu sunt doar asemănătoare – sunt identice. De exemplu, nu puteți defini o ancoră pentru un element și un ID pentru alt element, cum ar fi următoarele două, deoarece va apărea un conflict:

```
<A name="netscape" href="http://www.netscape.com">Netscape Home Page</A>
<IMG id="netscape" src="navigator.png">
```

Atât `name` cât și `id` ocupă același `namespace`, ceea ce înseamnă că sunt aceeași variabilă în domeniul de existență al documentului HTML în care se găsesc. Puteți folosi însă aceeași variabilă într-un document HTML diferit fără a crea un conflict, și aceasta deoarece ar fi în afara domeniului de existență al primului document.

SUGESTIE

Nu se face diferența între literele mari și mici din NAME și ID; deci Netscape, NETSCAPE și netscape sunt de fapt aceeași variabilă.

Deși specificația HTML oficială permite ancorelor să aibă doar un atribut, `name` sau `id`, acest lucru nu funcționează în practică. După cum am mai spus în secțiunea „Elementul ANCHOR", ancorele trebuie să aibă un atribut `href` pentru a fi recunoscute de Netscape Navigator sau de Internet Explorer.

Ancorele denumite sunt folosite în general pentru crearea legăturilor în cadrul documentului în care se află. Meniurile, de exemplu, pot fi create foarte ușor prin identificarea diferitelor secțiuni ale unei pagini Web cu ancore denumite și apoi prin aranjarea unor legături către adresele URL cu fragment la începutul paginii HTML. Această metodă este ilustrată în codul de exemplificare următor și în Figura 13.2:

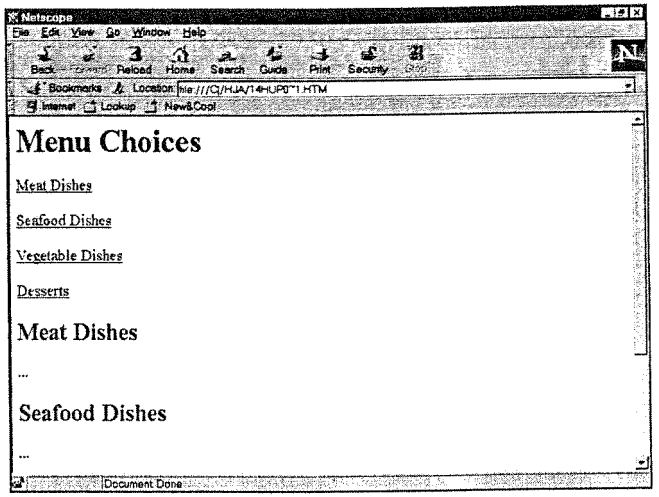
```
<H1>Menu Choices</H1>
<P><A href="#meat">Meat Dishes</A></P>
<P><A href="#fish">Seafood Dishes</A></P>
```



```
<P><A href="#veggies">Vegetable Dishes</A></P>
<P><A href="#fatcity">Desserts</A></P>
<H2 id="meat">Meat Dishes</H2>
...
<H2 id="#fish">Seafood Dishes</H2>
...
<H2 id="#veggies">Vegetable Dishes</H2>
...
<H2 id="#fatcity">Desserts</H2>
...
```

În acest exemplu, utilizatorii care selectează una dintre opțiunile meniului vor vedea acele secțiuni ale paginii afișate imediat, fără a mai fi nevoiți să desfășoare întrega pagină pentru a ajunge la ele. Observați că, în acest caz, ancorele denumite nu sunt realizate cu elementul ANCHOR și cu atributul `name`, ci cu atributele `id` ale mai multor elemente H2. Ca urmare, adresele URL cu fragment se referă la începutul secțiunii, iar primul rând de sus al rezultatului afișat este titlul secțiunii.

FIGURA 13.2.
Un meniu cu ancore denumite.



Ancorele denumite și adresele URL cu fragment pot fi de asemenea folosite pentru legături de la alte documente. Puteți ajunge din alt document la secțiunea dessert a meniului pe care tocmai l-am descris, prin folosirea unei legături către documentul meniului și adăugarea fragmentului, după cum se poate vedea:

```
<A href="http://www.eatfood.com/feedme.html#fatcity">Desserts</A>
```

Prin selectarea acestei legături, puteți accesa pagina referită. Afișajul începe cu pagina deja desfășurată în jos până în locul în care este ancora, în loc să pornească de la începutul paginii, ca de obicei.

Este posibil să avem un element ANCHOR care să îndeplinească un rol dublu, ca sursă de legătură și destinație a ancorei denumite. Trebuie doar să folosiți amândouă atributele `href` și `name` (sau `id`), ca în exemplul următor:

```
<A name="navigator" href="http://www.netscape.com/">Netscape Home Page</A>
```

Deoarece atributele name și id sunt echivalente, puteți scrie acest cod și în felul următor:

```
<A id="navigator" href="http://www.netscape.com/">Netscape Home Page</A>
```

Oricum scrieți, o legătură către ea ar avea codul următor:

```
<A href="#navigator">Netscape Link</A>
```

Legături imagine

Elementul ANCHOR permite folosirea imaginilor la crearea legăturilor nontextuale. Nu este necesară decât substituirea porțiunii de text a legăturii cu o adresă URL imagine. Puteți face acest lucru prin abordarea originală, din versiunile anterioare de HTML, ca în codul următor:

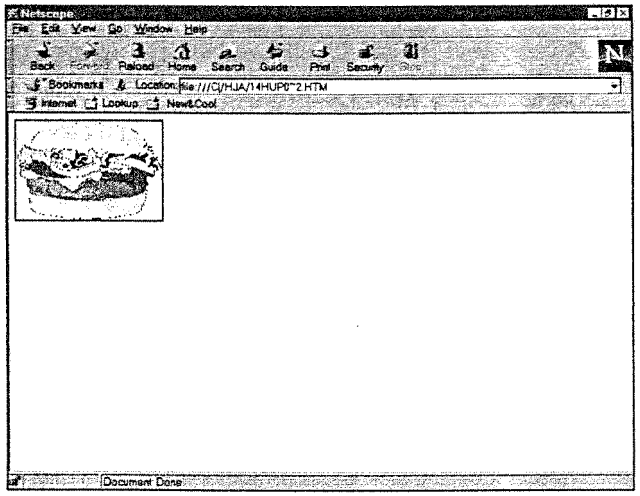
```
<A href="http://www.hamburger.com/index.html"><img src="hamburger.gif"></A>
```

Sau legătura poate fi codată printr-o metodă mai nouă, cu elementul OBJECT, ca în codul următor:

```
<A href="http://www.hamburger.com/index.html"><OBJECT data="hamburger.gif"
➥ type="image/gif"></A>
```

Indiferent de abordare, rezultatul final arată ca în Figura 13.3.

FIGURA 13.3.
O legătură imagine.



Puteți avea, de asemenea, atât text, cât și imagine în aceeași legătură, după cum se poate vedea în varianta ilustrată în Figura 13.4:

```
<A href="http://www.hamburger.com/index.html"><img src="hamburger.gif">
➥ Hamburgers</A>
```

De fapt, nu există nici un motiv tehnic pentru care să nu poată fi incluse mai multe imagini în aceeași legătură. Însă o legătură cu imagini multiple nu pare să ofere vreun avantaj deosebit, deoarece executarea unui clic pe oricare dintre aceste imagini conduce la același rezultat. Este totuși posibil ca acest tip de legătură să fie folositor, astfel încât vă prezentăm tehnica în codul următor, iar rezultatul în Figura 13.5:

```
<A href="http://www.hamburger.com/index.html"><img src="hamburger.gif">
➥ <img src="fries.gif">Burger and Fries</A>
```


FIGURA 13.4.
O legătură imagine cu text.

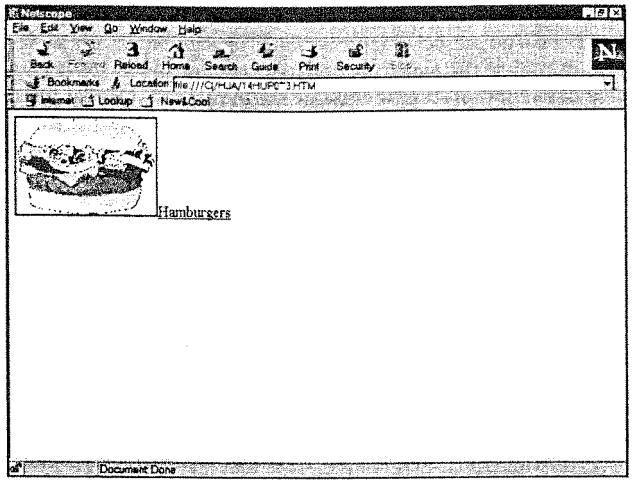
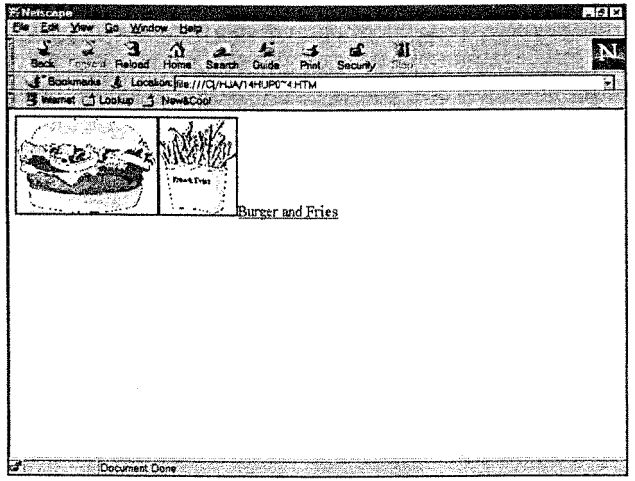


FIGURA 13.5.
O legătură imagine dublă cu text.



Legături către un conținut specializat

Unele legături către un conținut specializat sunt realizate prin aceleași metode ca legăturile către resursele HTML obișnuite. De exemplu, o legătură către un fișier Adobe Acrobat ar arăta în felul următor:

```
<A href="http://www.specialized.org/readme.pdf">README file in Acrobat format</A>
```

Totuși, pentru a citi fișierele Acrobat .pdf pe un browser Web, utilizatorul trebuie să aibă instalat cititorul Acrobat. Dacă acesta nu este deja instalat, atunci trebuie să-l descarce din rețea și să-l instaleze înainte de a putea accesa fișierul. Aduceți-vă aminte de posibilul consum suplimentar de timp și efort al utilizatorilor când decideți să folosiți fișiere nonstandard pe situl dumneavoastră Web. Această operație de descărcare-instalare, de

exemplu, ar putea întârzia accesul la fișierul dumneavoastră cu aproximativ o jumătate de oră – o presiune serioasă asupra răbdării unui surfer obișnuit de pe Web. Majoritatea oamenilor au deja instalate cele mai folosite plug-in-uri, dar poate ar fi mai bine să profitați de noile capacități oferite de foile de stil în cascadă, care vă permit să produceți o pagină Web cu aceeași precizie a controlului asupra fonturilor și poziționării ca și în cazul produselor externe. Folosirea foilor de stil va face ca vizitatorii sitului dumneavoastră să fie mai fericiți și, deoarece HTML-ul curat vă scutește de necesitatea utilizării unor produse și plug-in-uri externe, va reduce, de asemenea, costul și complexitatea producerii sitului dumneavoastră Web.

Cel mai simplu mod de a adăuga un fișier de sunet este să creați o hiperlegătură obișnuită către resursa de sunet, ca în codul următor:

```
<A href="themsong.wav">Play our theme song!</A>
```

Dezavantajul acestei abordări este că fișierul de sunet trebuie descărcat la client înainte de a fi rulat. În schimb, pentru un flux audio în timp real este necesar un software special, cum ar fi serverele RealAudio.

Fișierele de sunet RealAudio, spre deosebire de varianta legăturii directe, necesită o anumită metodă de invocare. În loc de o legătură directă către resursa de sunet, pagina dumneavoastră HTML se leagă printr-un metafișier special, care conține informații despre resursa de sunet. Aceste informații sunt apoi transmise înapoi browserului Web al utilizatorului, care le retransmite pentru procesare fie unui player autonom, fie unui plug-in.

Metafișierul conține doar adresa URL a resursei de sunet. Un metafișier RealAudio arată ca în exemplul următor:

```
pnm://www.yourserver.com/themesong.ra
```

Ca în orice schemă de URL, prima parte este un protocol (în acest caz, `pnm`, care este specific fișierelor RealAudio), urmat de adresa serverului și de calea către fișier. Fișierele RealAudio au mereu extensia `.ra`.

Pot fi stivuite împreună resurse multiple într-un singur metafișier, dacă doriți să fie rulate consecutiv, ca în exemplul următor:

```
pnm://www.yourserver.com/themesong.ra
pnm://www.yourserver.com/tada.ra
pnm://www.yourserver.com/introsong.ra
```

Metafișierul RealAudio nu conține informații de antet sau cod HTML. Deși pot fi adăugate alte date (cum sunt instrucțiunile de copyright), acestea sunt introduse în adresa URL propriu-zisă. Aceste metafișiere sunt salvate ca simple lișiere ASCII. Extensia unui metafișier depinde de aplicația care trebuie să-l redea: un plug-in sau un player autonom. Pentru plug-in-uri, extensia fișierului este `.rpm`, iar pentru playere individuale este `.ram`.

Nu aveți de unde să știți dacă vizitatorul sitului dumneavoastră are un plug-in sau un player individual, astfel încât ar trebui să salvați același metafișier cu amândouă extensiile de fișier. După cum probabil ați bănuit, există tot două metode de a invoca resursa de sunet, și trebuie să le folosiți pe amândouă pentru a acoperi toate posibilitățile.

Legătura pentru o apelare simplă a unei resurse de sunet RealAudio pentru un player individual este obișnuitul element `ANCHOR`, iar adresa URL pentru atributul `href` este adresa metafișierului. Un exemplu tipic ar arăta în felul acesta:

```
<A href="themesong.rpm">Play our theme song!</A>
```


În schimb, legătura pentru invocarea unui fișier RealAudio pentru un plug-in folosește elementul `EMBED`, care nu este standard pentru HTML. Codul următor va îngloba un obiect player RealAudio în pagina HTML:

```
<EMBED SRC="themesong.rpm" width=300 height=134>
```

Pentru a acoperi amândouă variantele, folosiți eticheta `NOEMBED` împreună cu celelalte două abordări discutate înainte:

```
<EMBED SRC="themesong.rpm" width=300 height=134>
<NOEMBED><A href="themesong.ram">Play our theme song!</A></NOEMBED>
```

Acest cod afișează un obiect player RealAudio în acele browsere care acceptă opțiunea și o legătură normală în cele care nu o acceptă.

O explicație completă despre instalarea și folosirea serverelor și fișierelor RealAudio depășește scopul acestei cărți; pentru mai multe informații, puteți găsi explicații complete pe situl Web al lui RealAudio, la `http://www.real.com/`.

Rezumat

Hiperlegăturile sunt cele mai importante elemente ale World Wide Web; fără ele, Webul nu ar putea exista. Sunt cele mai des folosite pentru a interconecta paginile HTML, în același sit sau pe situri diferite. De asemenea, sunt folosite pentru a adăuga imagini și alte obiecte unei pagini HTML.

Hiperlegăturile sunt bidirecționale, având atât o ancoră sursă, cât și o ancoră destinație. Majoritatea hiperlegăturilor sunt apelate cu eticheta `<A>`, reprezentând elementul `ANCHOR`, iar ancora destinație este specificată cu atributul `href` (referință hipertext), care este singurul atribut absolut necesar într-o legătură. Ancorele denumite sunt ancore destinație care pot fi folosite pentru a specifica hiperlegături către o anume secțiune a unei pagini HTML. Ele se compun fie dintr-un element `ANCHOR` care folosește atributul `name`, fie din alt element care folosește atributul `id`. Hiperlegăturile, de pe aceeași pagină sau de pe pagini diferite, pot face referire între ele cu o adresă URL cu fragment, această capacitate fiind des folosită la crearea meniurilor. Legăturile pot conține text, imagini sau amândouă. Hiperlegăturile pot fi, de asemenea, orientate către resurse non-HTML, cum ar fi fișierele Adobe Acrobat sau fișierele de sunet RealAudio. Unele resurse non-HTML folosesc exact aceeași sintaxă a legăturilor ca orice legătură HTML, dar altele necesită schimbări ale procedurii. Hiperlegăturile sunt de asemenea folosite în hărți imagine, care sunt discutate în capitolul următor.

Capitolul 14
Crearea meniurilor cu hărți imagine

de David și Rhonda Crowder

În acest capitol

- **Alegerea hărților imagine corespunzătoare 177**
- **Partea clientului sau partea serverului? 180**
- **Instrumente pentru harta imagine 186**

Probabil pentru că ne naștem cu capacitatea de a interpreta imaginile, pe când limbajele trebuie să le învățăm, hărțile imagine au un succes universal în fața meniurilor de tip text. Indiferent de motiv, hărțile imagine au devenit prima opțiune pentru dezvoltarea meniurilor pe World Wide Web.

Ce este exact o hartă imagine? Sigur vă amintiți din Capitolul 13, „Hiperlegături și ancore", că o hiperlegătură poate fi bazată pe imagine sau pe text. O *hartă imagine* este o versiune extinsă a unei hiperlegături imagine. În loc să avem o singură legătură posibilă indiferent unde executăm clic pe imagine, aceasta oferă mai multe legături diferite, în funcție de locul unde executați clic pe ea. O singură poză poate înlocui un meniu mare de tip text.

O ancoră imagine cu o singură legătură nu ridică dificultăți de procesare, dar hărțile imagine, cu hiperlegăturile lor multiple, necesită o metodă pentru a determina legătura care să fie urmată. Metoda stabilită a fost să se urmărească poziția indicatorului mouse-ului față de colțul din stânga-sus al imaginii. Acest colț are coordonatele 0,0 în notația cu x și y. Coordonata x, de la stânga la dreapta, crește pe măsură ce vă deplasați către dreapta; coordonata y, de sus în jos, crește pe măsură ce coborâți. (Vezi Figura 14.1.) Acestor coordonate, care, inițial, reprezentau doar forme rectangulare, le-a fost atașată o listă de legături. Dacă indicatorul este între 0,0 și 20,10, atunci este activată o hiperlegătură diferită decât dacă este între 0,11 și 20,20.

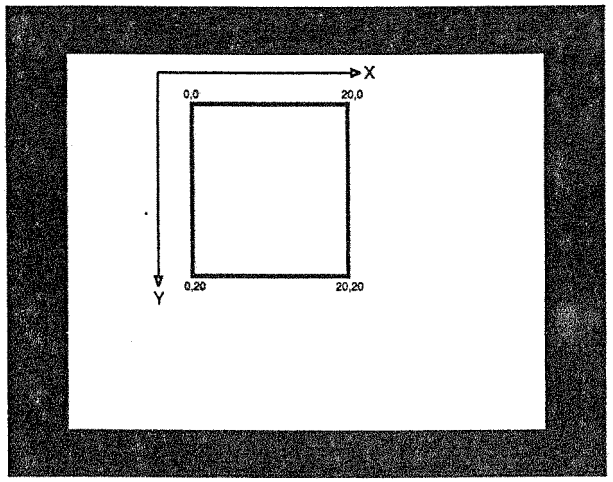


FIGURA 14.1.
Coordonatele hărții imagine.

Pe măsură ce s-a îmbunătățit tehnologia hărții imagine, au fost adăugate alte două forme ce ar putea fi folosite la codificarea hiperlegăturilor. Hărțile imagine moderne nu acceptă doar arii rectangulare, ci și circulare și poligonale.

ATRIBUTE COMUNE

Majoritatea elementelor din acest capitol au atribute comune, cum ar fi articolele care controlează identificarea, stilul, limbajul și tratarea evenimentelor. Deoarece atributele comune sunt atât de des folosite în HTML 4 (de aceea se numesc „comune"), veți găsi o prezentare a lor în introducerea acestei cărți. Puteți, de asemenea, găsi mai multe informații despre aceste des folosite caracteristici și elementele lor asociate în Anexa A.

Între timp, pe parcursul acestui capitol, ne vom referi la atributele comune folosind aceste cuvinte cheie: core (principale), international (internaționale) și event (evenimente).

Alegerea hărților imagine corespunzătoare

Desigur, nu puteți folosi chiar orice imagine veche fără să țineți cont de conținutul său; imaginile trebuie să aibă un sens ca hartă a unei hiperlegături. De exemplu, să spunem că ați fost angajat să creați un sit Web pentru o firmă de construcții. Clientul dorește ca situl să arate vederi din diferite camere ale unei case model. Metoda exactă este lăsată la latitudinea dumneavoastră.

Puteți folosi foarte simplu un meniu de tip text, ceva asemănător cu cel din Figura 14.2, și codul următor:

```
<P>Choose from the following links to view pictures of the rooms:
<P><A href="masterbedroom.html">Master Bedroom</A>
<P><A href="recroom.html">Recreation Room</A>
<P><A href="kitchen.html">Kitchen</A>
...
```

Acest design ar chema o nouă pagină pentru fiecare dintre camerele listate. Acele pagini probabil vor avea imagini înglobate ale camerei respective, urmate de o descriere a lor care ar putea include informații despre dimensiuni și opțiuni de mobilare.

Acest design este totuși primitiv, iar clientul probabil că nu va fi impresionat. Optând pentru un design mai elegant și mai grafic, decideți să construiți pagina într-o pereche de cadre. Cadrul de sus ar fi o hartă imagine; cel de jos va afișa imaginea camerei și informațiile despre ea.

Sunt mai multe imagini ale casei din care puteți alege, dar vederile din exterior nu se potrivesc scopului dumneavoastră deoarece din acestea nu pot fi selectate toate camerele. Printre materialele disponibile de la client, sunt și planurile podelei. Această soluție este perfectă pentru problema găsirii unei imagini care să vă permită să duplicați acel meniu de tip text, deoarece include o reprezentare vizuală a fiecărei camere din casă. Situl Web rezultat va arăta asemănător cu cel din Figura 14.3.

Acesta este un caz tipic de folosire a hărților reale pentru maparea imaginilor. Atunci când este vorba de o reprezentare geografică, este folosită o hartă standard. Pentru a arăta macheta casei, folosiți o hartă a casei. Dacă doriți să arătați distribuirea birourilor unei companii în Statele Unite, folosiți harta țării.

FIGURA 14.2.
Un meniu de tip text.

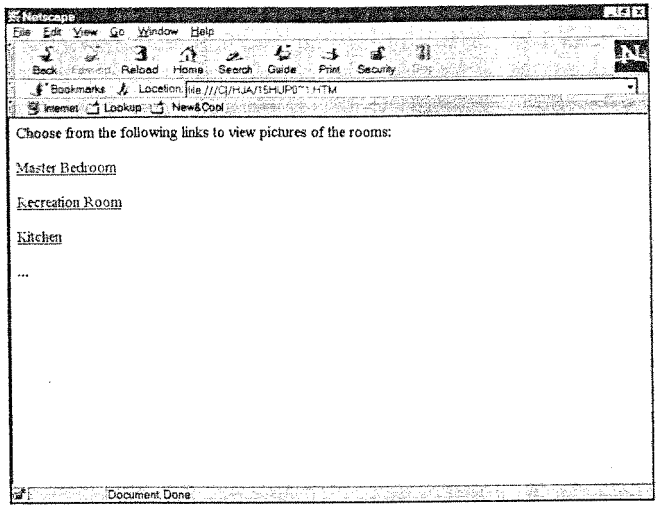
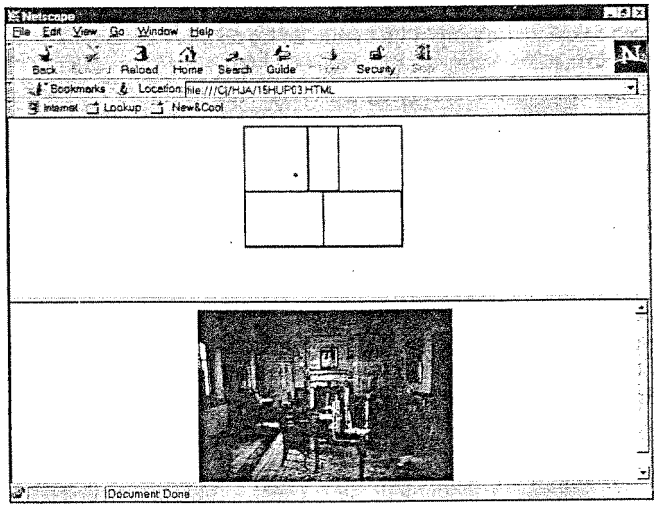


FIGURA 14.3.
Un meniu de tip hartă imagine.



Mai sunt însă și alte tipuri de „hărți", care conțin reprezentări vizuale ale elementelor constituente sau ale relațiilor dintre ele. Scopul unui meniu de tip hartă imagine este să furnizeze un set de legături ușor de înțeles către informațiile următoare. În acest sens, schema unui circuit electric este o hartă a componentelor sale. (Vezi Figura 14.4.) Fotografia unei farfurii cu mâncare pe o masă ar putea fi o hartă către diferite rețete, după cum se poate vedea în Figura 14.5, iar fotografia unei clase ar putea fi o hartă către informațiile biografice ale absolvenților. (Vezi Figura 14.6.)

Desigur, poate fi folosit și textul într-o hartă imagine, după cum se vede în Figura 14.7. Cuvintele sunt tratate ca o altă parte a imaginii.

FIGURA 14.4.
O hartă a componentelor.

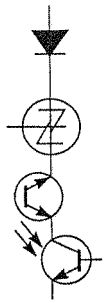


FIGURA 14.5.
O hartă de rețete.

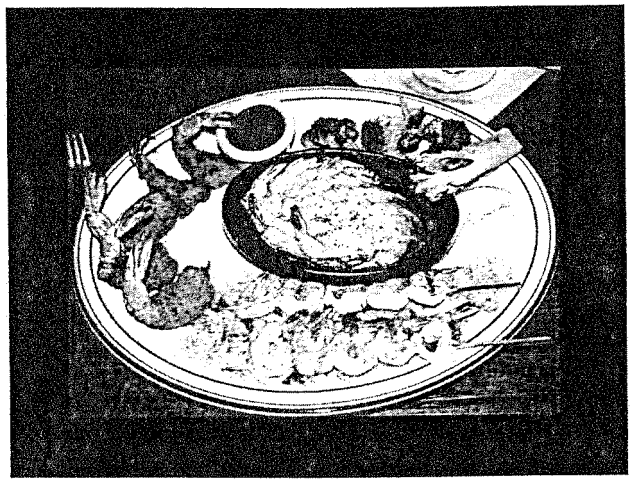


FIGURA 14.6.
O hartă biografică.

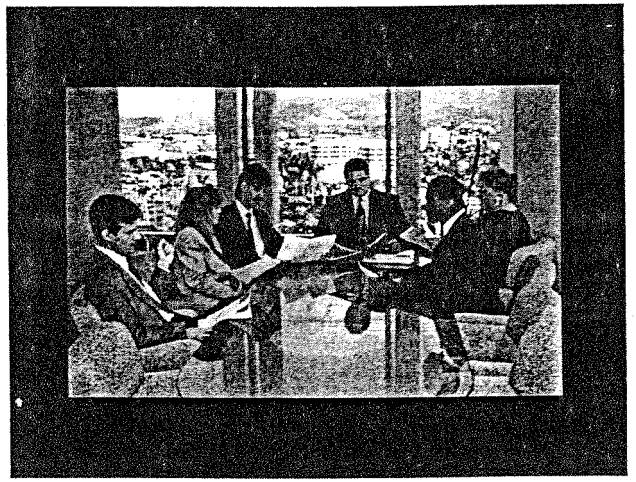
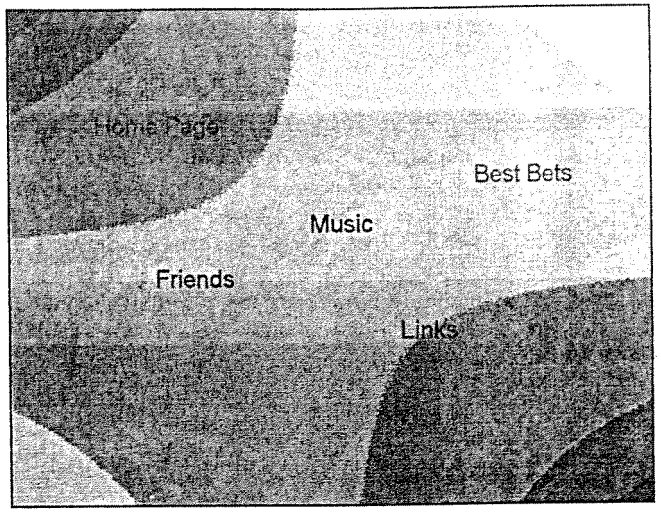


FIGURA 14.7.
O hartă cu text.



Partea clientului sau partea serverului?

După ce ați stabilit imaginea pentru harta dumneavoastră imagine, trebuie să alegeți metoda pe care o veți folosi pentru aceasta; hărțile imagine pot ține de client sau de server. Hărțile imagine ce țin de server sunt procesate de un program CGI pe serverul pe care se află situl Web, iar hărțile imagine ce țin de client sunt procesate de browserul Web sau alt agent al utilizatorului.

Cele de partea serverului, metoda veche, sunt într-un declin clar deoarece hărțile imagine ce țin de client au imensul avantaj de a comuta sarcina calculatorului pe sistemul individual al utilizatorului, în loc să o lase în seama serverului care, în cazul unui sit Web cunoscut, este deja încărcat peste măsură.

Hărțile imagine ce țin de client sunt de asemenea mai ușor de codat și oferă atât utilizatorului, cât și administratorului serverului avantaje funcționale. Când utilizatorul mută cursorul peste o hartă imagine ce ține de server, nu există nici o reacție. De cealaltă parte, hărțile imagine ce țin de client afișează adresa URL a legăturii în același fel ca o legătură ancoră de tip text, în linia de stare a browserului Web. Alt avantaj este că hărțile imagine de partea clientului pot folosi adrese URL parțiale, în timp ce hărțile imagine de partea serverului necesită adrese URL complete.

În ceea ce privește viteza, partea clientului câștigă detașat în fața părții serverului. Când un utilizator execută clic pe o hartă imagine ce ține de server, browserul Web trimite informațiile coordonatelor serverului HTTP, care le transmite unui program CGI care accesează informațiile adresei URL dintr-un fișier separat `.map`. După procesarea acestor date, programul CGI trimite adresa URL înapoi, prin server, browserului Web, care o retransmite apoi serverului sub forma unei cereri de hiperlegătură standard. Hărțile imagine ce țin de client sunt pur și simplu tratate în cadrul documentului HTML însuși. Nu este necesar nici un fișier extern sau accesul la server; prima oară când este implicat serverul HTTP în acest proces este când hiperlegătura este activată de browserul Web. În comparație cu greoaia metodă, a părții serverului, metoda HTML a părții clientului este un învingător detașat.

Hărți imagine ce țin de client

Primul lucru de care aveți nevoie pentru o hartă imagine ce ține de client este, bineînțeles, o imagine. Imaginea folosită pentru harta imagine este apoi introdusă în pagina HTML ca oricare alta, dar cu o diferență esențială: atributul `usemap`. Iată un exemplu de inserare a unei hărți imagine într-o pagină HTML:

```
<img src="sporting_goods.gif" usemap="#sportsmap">
```

Atributul `usemap` este cel care face posibilă folosirea imaginii ca hartă imagine. Nu numai că spune agentului clientului că aceasta este o hartă imagine, dar îl și direcționează către o ancoră denumită de pe pagina HTML unde pot fi găsite coordonatele și informațiile URL pentru această hartă imagine.

Acea ancoră denumită va fi un element `MAP`, care de obicei conține același meniu de tip text pe care l-ar conține pagina HTML dacă ați fi procedat după metodele vechi. Este o listă de legături, dar acestor legături li s-a adăugat ceva: o definire a ariei din harta imagine care le va activa.

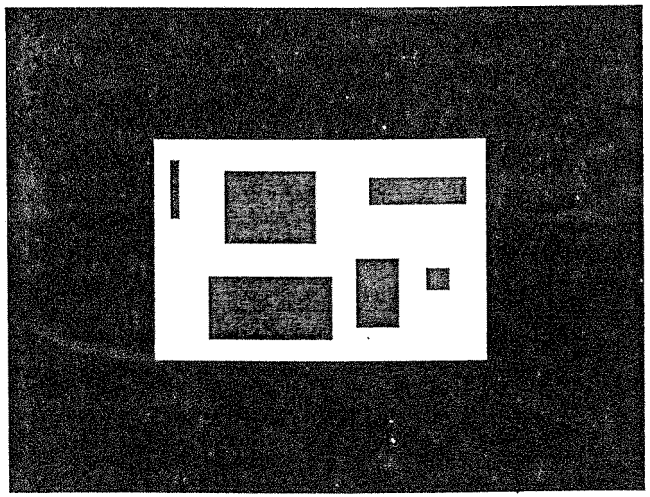
Elementele `MAP`, conținute în elementul `BODY`, au un singur atribut, `name`. Atributul `name` este folosit la identificarea unui anume element `MAP`, astfel încât acesta să poată fi accesat prin atributul `usemap` al elementului `IMG`. Numai un singur articol este permis între etichetele de început și de sfârșit ale elementului `MAP`: elementul `AREA`.

Elementul `AREA` este locul în care se găsesc legăturile și coordonatele reale. Întâi definește forma ariei din imagine, apoi furnizează coordonatele și după aceea specifică adresa URL a hiperlegăturii care va fi activată dacă se execută un clic în timp ce indicatorul mouse-ului se află în aria respectivă. Sunt trei forme diferite care pot fi definite cu atributul `shape`: `rect`, `circle` și `poly`. Ele se referă la formele dreptunghice, rotunde și poligonale. Dacă nu specificați o formă, atributul prestabilit este la valoarea `rect`.

Dreptunghiurile (vezi Figura 14.8) necesită două perechi de coordonate. Prima pereche reprezintă colțul din stânga-sus, iar a doua pe cel din dreapta-jos:

```
<AREA shape=rect coords="upper_left_x,upper_left_y, lower_right_x,lower_right_y"
➥ href="hyperlink_URL">
```

FIGURA 14.8.
Forme dreptunghice.



Cercurile necesită o pereche de coordonate – care furnizează coordonatele punctului central – și un al treilea număr care furnizează raza cercului:

```
<AREA shape=circle coords="center_x,center_y, radius" href="hyperlink_URL">
```

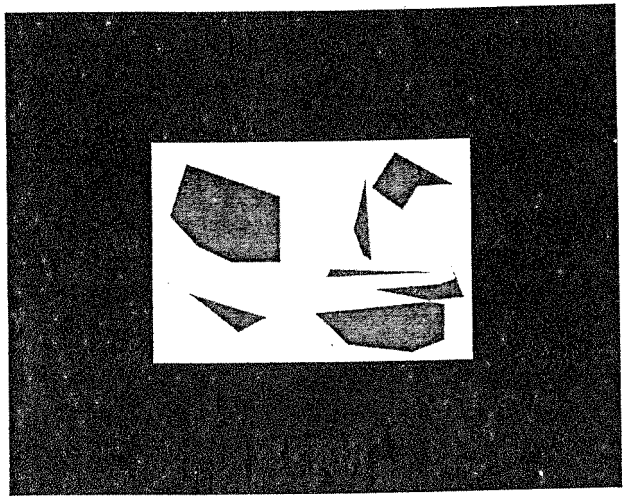
Poligoanele (vezi Figura 14.9), prin definiție forme mult mai complexe decât cele dreptunghice sau rotunde, necesită o serie de perechi de coordonate care definesc puncte de-a lungul marginilor figurii:

```
<AREA shape=poly coords="first_point_x, first_point_y,second_point_x,
➥ second_point_y,..., last_point_x,last_point_y"  href="hyperlink_URL">
```

OBSERVAȚIE

Probabil știți că poligonul este o formă geometrică cu mai multe laturi. În acest sens, putem spune că dreptunghiul este un poligon. În terminologia hărților imagine, poligoanele sunt totuși puțin diferite. Este adevărat că puteți folosi un poligon pentru a defini o formă dreptunghică, dar nu aveți nevoie de acest lucru, deoarece există deja o unealtă specială pentru acest scop. Poligoanele sunt de obicei folosite pentru a crea o serie de linii scurte, care se închid formând o formă mai ciudată care nu poate fi ușor inclusă într-un cerc sau într-un dreptunghi.

FIGURA 14.9.
Forme poligonale.



Atributul `shape` are și o a patra valoare posibilă, dar aceasta nu este cu adevărat o formă. Această valoare este `default`, care stabilește adresa URL care va fi activată când utilizatorul execută un clic înăuntrul hărții imagine, dar în afara oricărei arii.

ATENȚIE!

Valoarea prestabilită a atributului `shape` este `rect`, nu `default`. Denumirea valorii default (prestabilit) este doar o alegere nefericită.

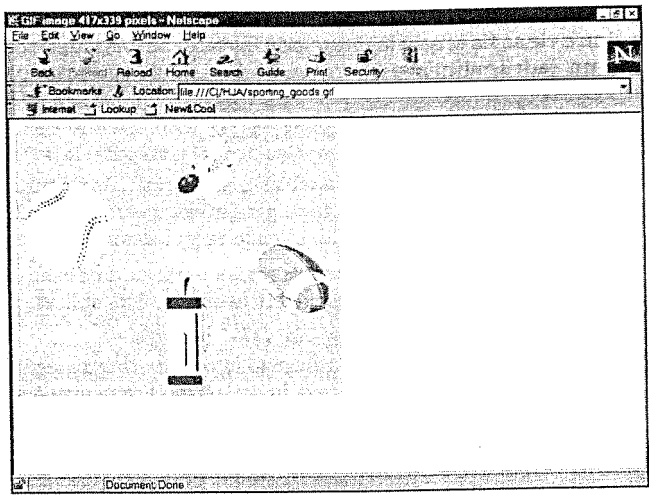
Următorul exemplu arată elementul `MAP` pentru harta imagine din Figura 14.10. Adresele URL au fost numite pentru a arăta cărui echipament sportiv corespund în harta imagine, astfel încât să puteți compara codul cu figura:

```
<MAP name="sportsmap">
<AREA shape=circle coords="65,129,57"
➥ href="http://www.sportgoods.com/baseball.html">
```



```
<AREA shape=rect coords="185,186,251,328"
➥ href="http://www.sportgoods.com/golf.html">
<AREA shape=poly coords="396,231,403,219,405,202, 400,188,390,175,376,162,357,
➥ 152,337,151,324,153, 317,157,314,162,309,170,309,184,317,204,326,216,337,224,
➥ 348,232,358,234,374,237,389,234,398,226,398,228,396,231"
➥ href="http://www.sportgoods.com/football.html">
<AREA shape=poly coords="183,94,257,96,300,42,270,11,245,11,215,39, 215,53,177,
➥ 53,169,72,148,93,181,95,184,98,183,94"
➥ href="http://www.sportgoods.com/bowling.html">
<AREA shape=default href="http://www.sportgoods.com/index.html">
</MAP>
```

FIGURA 14.10.
O hartă imagine sportivă.



Forma rotundă, bineînțeles că a fost aleasă pentru mingea de baseball. Datorită formelor neregulate ale multor imagini, o potrivire atât de perfectă nu este prea des posibilă. Deși trusa de golf nu este o formă dreptunghică perfectă, conturul său poate fi închis complet cu un dreptunghi, ceea ce este mult mai ușor decât folosirea unui poligon. Dacă un utilizator ar executa un clic foarte aproape, dar nu chiar pe trusa de golf, efectul ar fi același ca și cum clicul ar fi fost executat chiar pe ea, deoarece dreptunghiul este puțin mai mare decât trusa, cuprinzând mânerul și crosele. A trebuit ca imaginile pentru fotbal american și bowling să fie realizate folosind poligoanele, deoarece un cerc sau un dreptunghi nu ar fi putut, nici măcar aproximativ, să le cuprindă corespunzător. Dintre acestea două, mingea de fotbal american are cele mai multe perechi de coordonate, deoarece forma sa a fost urmărită cât mai de aproape de către contur. Poligonul în jurul scenei de bowling este mult mai aproximativ, dar totuși funcțional.

OBSERVAȚIE

La fel ca în cazul imaginilor obișnuite, o hartă imagine mare poate mări considerabil timpul de descărcare. Este imposibil de fixat o limită arbitrară a mărimii pentru toate hărțile imagine, dar ar trebui să folosiți cele mai mici imagini care sunt ușor de citit și de folosit. Harta imagine sportivă folosește o imagine 417X339 și, probabil, nu ar trebui să folosiți una mai mare.

După cum puteți vedea, problema majoră în dezvoltarea elementului `MAP` este stabilirea coordonatelor (fără să mai spunem de introducerea lor fără greșeli!). O metodă de a face

acest lucru este să încărcați imaginea pe care vreți să o folosiți pentru harta imagine într-un program de grafică, majoritatea acestora afișând coordonatele indicatorului. Apoi, în timp ce mutați indicatorul de-a lungul marginilor, notați coordonatele punctelor critice. Când terminați, introduceți numerele scrise de dumneavoastră. Totuși, această metodă este obositoare, frustrantă și supusă unei mulțimi de greșeli posibile, fiecare dintre ele putând zădărnici toată munca dumneavoastră.

Soluția la care au apelat majoritatea autorilor de cod HTML este folosirea programelor special concepute pentru construirea hărților imagine. În acest fel am realizat și noi exemplul. Aceste programe sunt explorate în detaliu în secțiunea „Instrumente pentru harta imagine".

Elementul AREA are câteva atribute foarte interesante. Dacă folosiți cadre, puteți specifica în care dintre ele să fie afișată legătura prin folosirea atributului target. De exemplu, dacă ați stabilit un cadru intitulat link_frame, atunci puteți specifica o pagină legată pentru a fi încărcată în el prin folosirea codului următor:

```
<AREA shape=rect coords="185,186,251,328"
➥ href="http://www.sportgoods.com/golf.html" target="link_frame">
```

Aveți mai multe opțiuni diferite când folosiți cadrul țintă. Atributul target din elementul AREA funcționează în același fel ca atributul target din elementul BASE. Pentru o explicare completă a atributului target, vezi Capitolul 12, „Să înțelegem locatorii uniformi de resurse".

Atributul nohref este disponibil pentru acele situații ciudate când doriți să definiți o formă, dar să preveniți formarea unei legături în cadrul ei. Este aplicat în felul următor:

```
<AREA shape=rect coords="185,186,251,328" nohref>
```

Atributul alt este folosit pentru a furniza informații despre legătură pentru agenții non-grafici ai utilizatorilor și funcționează în același mod ca pentru o imagine obișnuită cu elementul IMG (vezi Capitolul 11, „Adăugarea imaginilor pe pagina dumneavoastră Web".)

Elementul AREA acceptă, de asemenea, atributele pentru folosirea tastaturii tabindex și accesskey, care sunt aprofundate în Capitolul 15 și în Capitolul 41.

SUGESTIE

Dacă două arii diferite se suprapun, iar un utilizator execută un clic în partea comună, atunci va fi activată legătura ariei exprimată prima în elementul MAP. Aceasta deoarece browserul Web citește listarea de sus în jos, căutând un element AREA care să conțină coordonatele la care s-a executat clicul; după ce a găsit primele coordonate care se potrivesc, oprește căutarea și nu le va mai încerca și pe celelalte.

Metoda obiectului hartă imagine

HTML 4.0 a introdus o abordare alternativă pentru adăugarea hărților imagine ce țin de client în paginile Web. Acestea pot fi codate cu elementul OBJECT în locul elementului MAP. Această metodă nu folosește un element IMG separat cu un atribut usemap care să fie orientat către elementul MAP. În loc de aceasta, elementele AREA sunt conținute doar în cadrul elementului OBJECT însuși.

Următorul exemplu arată aceeași hartă imagine folosită în exemplul precedent, dar codată sub forma unui OBJECT în locul unei combinații de elemente IMG și MAP:

```
<OBJECT data="sporting_goods.gif" shapes>
<AREA shape=circle coords="65, 129, 57"
➥ href="http://www.sportgoods.com/baseball.html">
<AREA shape=rect cards="185,186,251,328"
➥ href="http://www.sportgoods.com/golf.html">
```



```
<AREA shape=poly coords="396,231,403,219,405,202,400,188,390,175,376,162,357,
➥ 152,337,151,324,153,317,157,314,162,309, 170,309,184,317,204,326,216,
➥ 337,224,348,232,358, 234,374,237,389,234,398,226,398,228,396,231"
➥ href="http://www.sportgoods.com/football.html">
<AREA shape=poly coords="183,94,257,96,300,42,270, 11,245,11,215,39,215,53,177,
➥ 53,169,72,148,93,181,95,184,98,183,94"
➥ href="http://www.sportgoods.com/bowling.html">
<AREA shape=default href="http://www.sportgoods.com/index.html">
<OBJECT>
```

Inima sistemului sunt elementele AREA, care sunt folosite în același fel în amândouă abordările. Întreaga idee de a transforma un element OBJECT într-o hartă imagine stă în atributul shapes din eticheta de deschidere, care comunică analizatorului HTML că vor urma coordonatele hărții imagine.

ATENȚIE!

Fiți atent și rețineți că atributul pentru autodefinirea elementului OBJECT ca hartă imagine este shapes (cu s la sfârșit – plural); nu este același lucru cu atributul elementului AREA, shape (fără s la sfârșit – singular), care este folosit pentru a specifica ce fel de formă (rect, circle sau poly) este aria.

Hărți imagine ce țin de server

Hărțile imagine ce țin de server necesită un fișier extern pentru a păstra coordonatele hărții. Acesta este numit, după cum vă puteți imagina, fișier .map. Totuși, fișierele .map nu sunt atât de simple pe cât ar putea fi. Trebuie să determinați dacă serverul dumneavoastră HTTP se conformează standardelor CERN sau NCSA pentru hărți imagine. Amândouă tipurile de servere lucrează cu fișiere .map, dar fiecare necesită un format al acestora puțin diferit față de celălalt și fiecare folosește un program diferit pentru a lucra cu ele.

Fișierul .map CERN pentru exemplul anterior ar arăta în felul următor:

```
default http://www.sportgoods.com/index.html
circ (65,129) 57 http://www.sportgoods.com/baseball.html
rectangle (185,186) (251,328) http://www.sportgoods.com/golf.html
polygon (396,231) (403,219) (405,202) (400,188) (390,175) (376,162) (357,152)
➥ (337,151) (324,153) (317,157) (314,162) (309,170) (309,184) (317,204)
➥ (326,216)  (337,224) (348,232) (358,234) (374,237) (389,234) (398,226)
➥ (398,228) (396,231) http://www.sportgoods.com/football.html
polygon (183,94) (257,96) (300,42) (270,11) (245,11) (215,39) (215,53)
➥ (177,53) (169,72) (148,93) (181,95) (184,98) (183,94)
➥ http://www.sportgoods.com/bowling.html
```

Observați că, spre deosebire de ordinea lucrurilor în elementul HTML AREA, formatul CERN listează întâi forma, apoi perechile de coordonate, fiecare între paranteze și apoi adresa URL a hiperlegăturii. Mai este încă o diferență în modul în care sunt descrise formele în sine: „circle" devine „circ", „rect" devine „rectangle" și „poly" devine „polygon", dar „default" rămâne „default". Coordonatele pentru forma circulară sunt separate de rază, care nu este închisă între paranteze.

Fișierul .map NCSA pentru aceeași mostră de imagine ar arăta în felul următor:

```
default http://www.sportgoods.com/index.html
circ http://www.sportgoods.com/baseball.html 65,129 65,186
rect http://www.sportgoods.com/golf.html 185,186 251,328
poly http://www.sportgoods.com/football.html 396,231 403,219 405,202 400,188
➥ 390,175 376,162 357,152 337,151 324,153 317,157 314,162 309,170 309,184
➥ 317,204 326,216 337,224 348,232 358,234 374,237 389,234 398,226 398,228 396,231
```

```
poly http://www.sportgoods.com/bowling.html 183,94 257,96 300,42 270,11 245,11
➥ 215,39 215,53 177,53 169,72 148,93 181,95 184,98 183,94
```

Observați că este folosită aceeași terminologie ca în elementul HTML AREA. Așadar „circle" este „circle", „rect" este „rect", „poly" este „poly" și „default" este „default". Singura diferență reală este că perechile de coordonate sunt separate prin spații. Forma circulară este definită de punctul central și un punct de pe circumferință.

În fiecare dintre aceste metode este necesară folosirea elementului IMG; pentru hărțile imagine ce țin de server nu există abordarea prin elementul OBJECT, deoarece reprezintă alt nivel al tehnologiei. Totuși, nu este permisă folosirea atributului usemap în acest caz. Este folosit în locul său atributul mai vechi ismap, ca în exemplul următor:

```
<A href= http://www.sportgoods.com/cgi-bin/htimage/sports.map>
➥ <img src="sporting_goods.gif" ismap></A>
```

Acest cod invocă handlerul încorporat în CERN pentru harta imagine, care este denumit htimage. Pentru a folosi handlerul NCSA pentru harta imagine, numit imagemap, ar trebui făcută următoarea modificare:

```
<A href= http://www.sportgoods.com/cgi-bin/imagemap/sports.map>
➥ <img src="sporting_goods.gif" ismap></A>
```

Amestecarea părții clientului cu cea a serverului

Dacă vreți să aveți suport pentru amândouă tipurile de hărți imagine, acest lucru este posibil. Desigur, necesită o muncă suplimentară. Trebuie să creați amândouă tipurile de date de hartă: un fișier .map și un element MAP. O dată ce ați realizat acest lucru, tot ce mai aveți de făcut este să codați legătura hărții imagine în același fel ca pentru abordarea de partea serverului și apoi să adăugați atributul usemap.

Tehnica hărții imagine dublă cu un server CERN arată în felul următor:

```
<A href= http://www.sportgoods.com/cgi-bin/htimage/sports.map>
➥ <img src="sporting_goods.gif" ismap usemap="#sportsmap"></A>
```

Aceeași tehnică cu un server NCSA arată în felul următor:

```
<A href= http://www.sportgoods.com/cgi-bin/imagemap/sports.map>
➥ <img src="sporting_goods.gif" ismap usemap="#sportsmap"></A>
```

Instrumente pentru harta imagine

După cum am mai spus, determinarea și codarea perechilor de coordonate pentru punctele care definesc forma într-o hartă imagine este o sarcină intimidantă. Din fericire, această problemă a fost rezolvată cu ajutorul unor programe specializate, cu interfețe care permit autorilor de cod HTML să deseneze zone pe care se poate executa clic chiar pe imagine. Aceste programe translatează apoi acele linii desenate în codul necesar implementării hărții imagine și vă permit să le salvați în orice format de fișier pentru hărți imagine.

Harta imagine sportivă din exemplul precedent (vezi Figura 14.10) a fost construită prin folosirea unui program freeware numit Map This!. Mai multe astfel de programe sunt listate în Anexa G.

Pentru început, trebuie să aveți deja o imagine (în formatul GIF sau JPEG) pe care intenționați să o transformați într-o hartă imagine.

Pentru a încărca o imagine în Map This!, apăsați butonul New File (sau alegeți File | New din meniu) pentru a deschide caseta de dialog Make New Image Map (vezi Figura 14.11), care doar vă amintește să folosiți GIF-uri sau JPEG-uri deja existente. Dacă nu vreți să vă apară această casetă de dialog ori de câte ori încărcați un nou fișier imagine, selectați caseta

de validare Don't show this message.... Executați un clic pe butonul Okay pentru a merge mai departe.

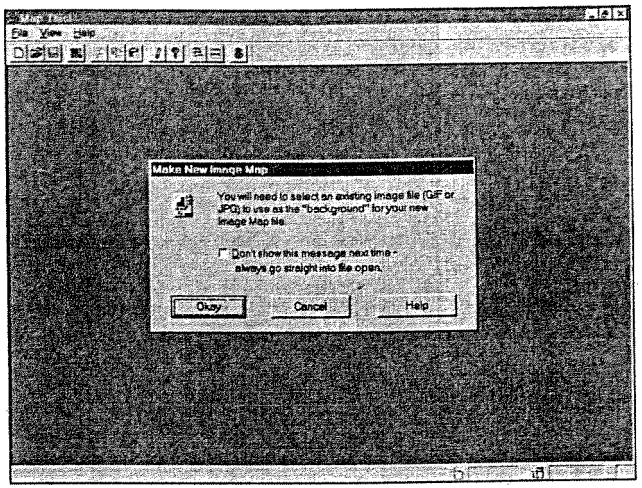


FIGURA 14.11. *Caseta de dialog Make New Image Map.*

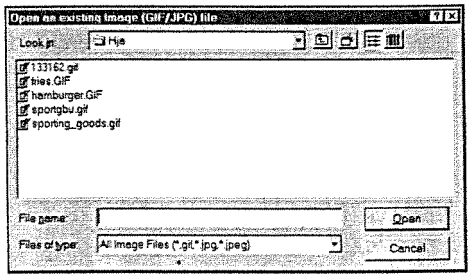


FIGURA 14.12. *Deschiderea unui fișier imagine în Map This!*

Următorul lucru care apare este o casetă de dialog standard pentru deschiderea fișierelor (vezi Figura 14.12), care este setată să arate numai fișierele GIF și JPG (sau JPEG). Selectați numele fișierului pe care vreți să-l încărcați pentru harta imagine a dumneavoastră. În acest exemplu, vom folosi sporting_goods.gif. Apăsați butonul Okay (puteți executa un dublu clic pe numele fișierului în loc să selectați fișierul și apoi să apăsați butonul Open).

Fișierul pe care îl veți folosi este acum încărcat. Executați clic pe butonul de maximizare din colțul din drepta-sus al ferestrei care conține imaginea pentru a o aduce la mărimea maximă, după cum este arătat în Figura 14.13.

Pe bara cu instrumente există câte un buton pentru fiecare dintre cele trei forme: dreptunghi, cerc și poligon. Alt buton, lângă cele de formă, permite selectarea ariilor după ce le-ați creat.

Pentru a seta aria hărții imagine pentru baseball, apăsați butonul Circle, apoi mutați indicatorul în centrul mingii de baseball. Țineți apăsat butonul din stânga al mouse-ului și mutați indicatorul până când cercul rezultat este de mărimea mingii. Nu contează dacă nu este perfect centrat. Apoi, apăsați butonul Select și mutați indicatorul în orice loc din cercul albastru. În timp ce țineți apăsat butonul mouse-ului, trageți cercul până când acoperă în totalitate mingea de baseball. (Vezi Figura 14.14) Cercul albastru se schimbă într-o arie hașurată în timpul acestei operațiuni, ceea ce este normal pentru oricare dintre ariile selectate. Dați drumul butonului de mouse.

Butonul Polygon Butonul Select

FIGURA 14.13.
Imaginea extinsă.

Butonul Rectangle
Butonul Circle

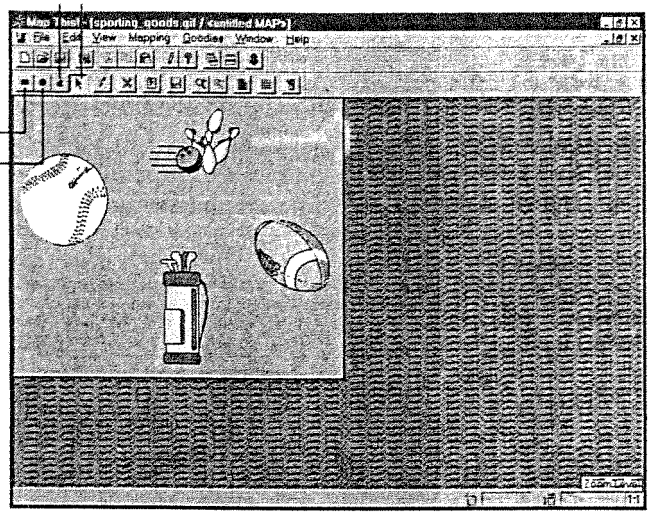
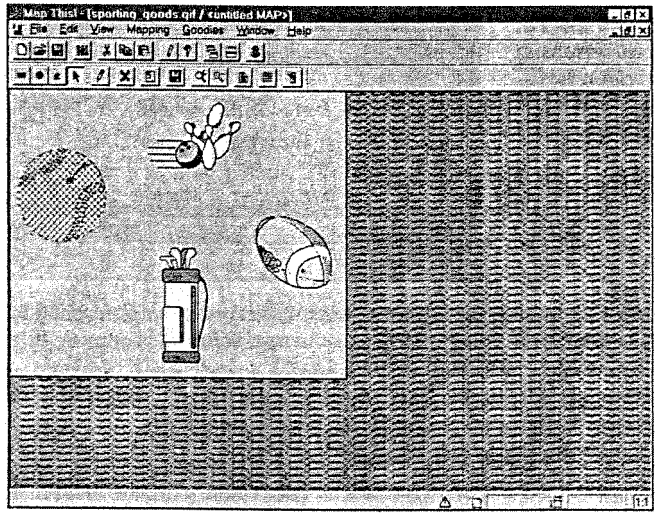


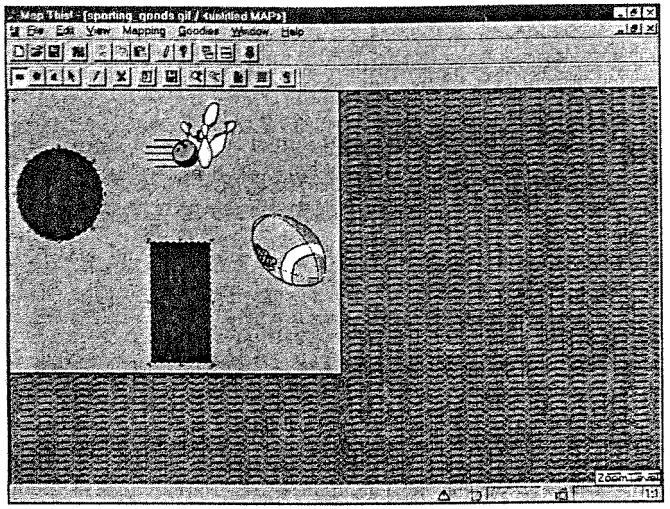
FIGURA 14.14.
Mingea de baseball acoperită.



Dacă mărimea nu este perfectă, aveți două opțiuni. Prima este să ignorați acest lucru. Majoritatea utilizatorilor execută clicul în centrul legăturii hartă imagine, așadar probabil că nu o să conteze prea mult dacă mărimea este un pic diferită de imagine. Totuși, dacă doriți să o ajustați, mutați indicatorul până când este deasupra unui mâner de prindere dintre cele aflate pe periferia cercului. Când indicatorul se shimbă într-o săgeată dublă, țineți apăsat butonul din stânga al mouse-ului și mutați indicatorul înăuntru sau în afară până când cercul are exact mărimea dorită. Dacă este necesar, repetați operațiunea. Folosiți tehnica de selectare pe care tocmai ați învățat-o pentru a repoziționa aria, dacă este nevoie.

Apoi, apăsați butonul Rectangle. Plasați indicatorul mouse-ului în colțul din stânga-jos al trusei de golf. Țineți apăsat butonul mouse-ului și mutați indicatorul în sus și în dreapta până când toată trusa, inclusiv mânerul, este cuprinsă în dreptunghi. (Vezi Figura 14.15.) Dacă aveți nevoie să schimbați mărimea dreptunghiului, folosiți aceeași metodă ca în cazul cercului. Singura diferență este că mânerele de prindere afectează doar partea pe care se află; dacă sunt într-un colț, vor afecta doar cele două laturi care se întâlnesc în acel colț.

FIGURA 14.15.
Trusa de golf acoperită.



OBSERVAȚIE

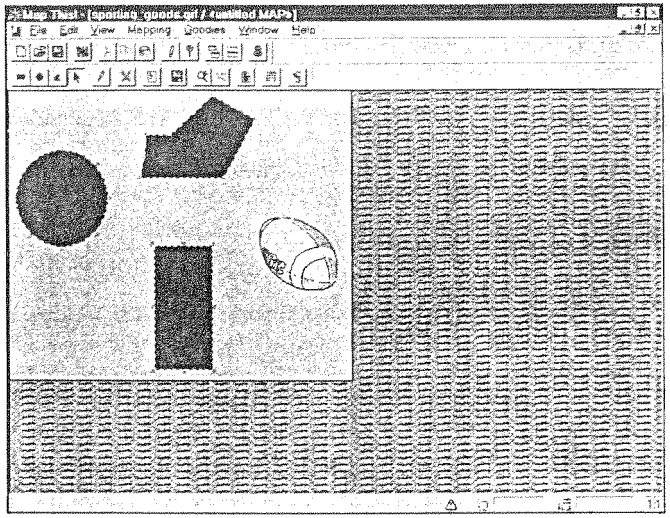
Am ales colțul din stânga-jos al trusei de golf ca punct de plecare în acest exemplu deoarece era un punct de referință mai clar decât cele de pe partea dreaptă a trusei, pe care se află mânerul care iese în afară. Totuși, puteți începe un dreptunghi în orice punct și să mutați indicatorul mouse-ului în orice direcție; nu este obligatoriu să mergeți din stânga-jos în dreapta-sus.

Următoarea formă pe care o vom folosi este poligonul. Desenarea unui poligon este puțin mai complexă decât a unui cerc sau a unui dreptunghi, dar constă într-un singur pas, repetat de mai multe ori. Procedura este să executați un clic cu butonul stâng al mouse-ului pentru a vă alege un punct, să mutați indicatorul în următorul loc în care doriți să plasați un punct, să executați alt clic, să mutați din nou și așa mai departe. Pe măsură ce faceți acest lucru o linie mobilă vă urmărește acțiunile, un capăt fiind ancorat în ultimul punct stabilit, iar celălalt capăt în poziția curentă a indicatorului. Poate că sună complicat, dar de fapt este foarte simplu de realizat.

Apăsați butonul Polygon, apoi plasați indicatorul mouse-ului în colțul din stânga-jos al scenei de bowling. Mutați indicatorul către dreapta (fără să țineți butonul apăsat) până când ajungeți într-un punct din dreapta scenei de bowling și apăsați butonul stâng al mouse-ului pentru a stabili un punct aici. Apoi, mutați indicatorul oblic către dreapta-sus până când depășiți popicul înclinat către dreapta, executați alt clic, mutați înspre stânga-sus până când ajungeți deasupra celui mai înălțat popic și executați clic iarăși. Acum, mutați indicatorul către stânga-jos până când atingeți un punct superior liniilor orizontale, executați din nou un clic și mutați indicatorul drept către stânga până ajungeți într-un

punct de deasupra punctului de pornire. Executați un dublu clic pe butonul din stânga al mouse-ului pentru a completa poligonul. Figura 14.16 arată poligonul obținut.

FIGURA 14.16.
Scena de bowling selectată.



OBSERVAȚIE

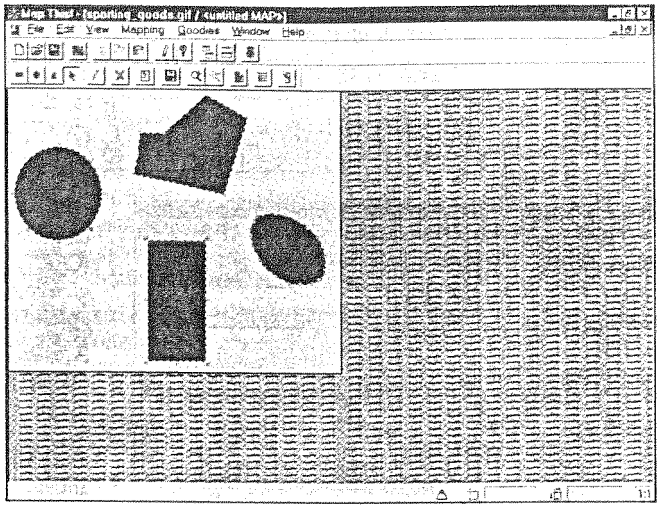
Deși Map This! adaugă automat ultima linie conectând punctul terminal cu cel de început, aceasta este doar o atenție drăguță. Un poligon nu trebuie să fie complet închis pentru a defini o arie. Browserul Web sau alt agent al clientului conectează automat primul punct cu ultimul pentru a-l închide. Așadar, un poligon necesită, de fapt, un minim de trei puncte pentru a închide complet aria dintre cele două linii pe care le formează (desigur, acest număr minim de puncte creează doar triunghiuri; veți avea nevoie să adăugați mai multe puncte pentru o formă mai complexă).

Schimbarea dimensiunii unui poligon, într-o exprimare strictă, este imposibilă. Totuși, puteți muta punctele unul câte unul. Făcând acest lucru, afectați forma poligonului, putând fi ajustată astfel încât să cuprindă mai exact forma obiectului vizat, dacă acest lucru este necesar. Această metodă funcționează în mare parte asemănător cu mânerele de prindere ale dreptunghiurilor, cu excepția că indicatorul se schimbă într-o cruce în locul unei săgeți duble. Ținând apăsat butonul stâng al mouse-ului și mutând punctele respective, afectați doar cele două linii ancorate în acel punct. Mutarea întregului poligon, spre deosebire de schimbarea formei sale, se face în același fel ca la celelalte două forme.

În cele din urmă, aplicați o formă poligonală mingii de fotbal. Spre deosebire de scena de bowling, pe care ați urmat-o doar aproximativ, urmăriți cu atenție conturul acestei forme. După ce ați apăsat butonul Polygon, plasați indicatorul mouse-ului la unul dintre vârfurile mingii de fotbal și executați un clic cu butonul stâng pentru a fixa un punct în acel loc. Acum, mutați indicatorul de-a lungul marginii mingii până când linia dintre primul punct și indicatorul mouse-ului se potrivește cât mai bine cu marginea mingii. Dacă vă duceți prea departe, linia dreaptă va deveni prea lungă și nu va urma conturul mingii. Dacă se întâmplă acest lucru, pur și simplu mutați indicatorul mouse-ului înapoi până când linia se potrivește mai bine, apoi apăsați butonul mouse-ului. Continuați să faceți acest lucru până când ați înconjurat complet mingea de fotbal (care va necesita multe puncte), apoi executați

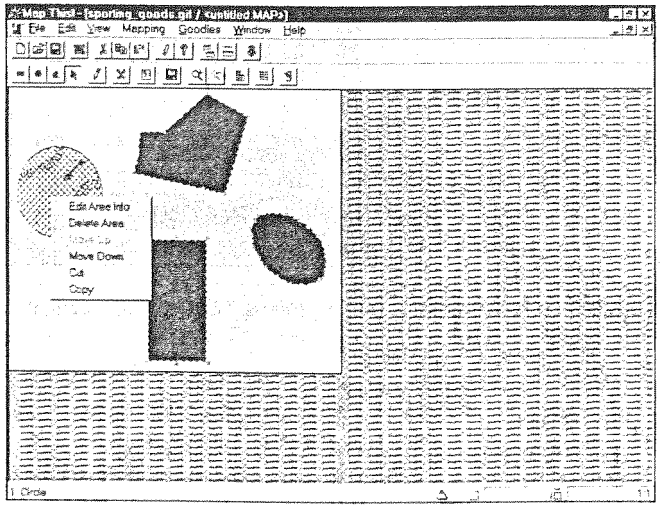
un clic dublu cu butonul stâng al mouse-ului. Rezultatul ar trebui să fie ceva asemănător celui din Figura 14.17.

FIGURA 14.17.
Mingea de fotbal acoperită.



Acum trebuie să adăugați adresele URL pentru fiecare arie. Apăsați butonul Select și apoi mingea de baseball. Apăsați butonul drept al mouse-ului pentru a afișa meniul pop-up (vezi Figura 14.18) și alegeți Edit Area Info pentru a deschide caseta de dialog Area Settings. (Vezi Figura 14.19.) Introduceți adresa URL care va fi activată pentru această arie și orice alt comentariu pe care ați vrea să-l includeți, apoi apăsați butonul Ok. Repetați procedura pentru toate celelalte forme.

FIGURA 14.18.
Un meniu pop-up.



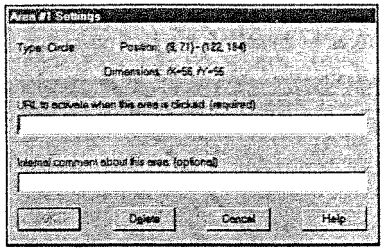


FIGURA 14.19. *Caseta de dialog Area Settings.*

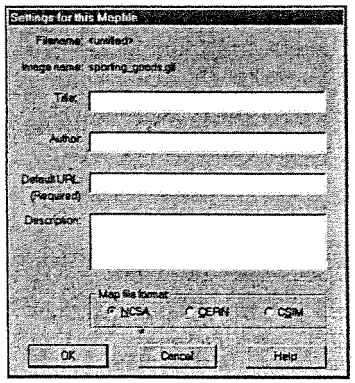


FIGURA 14.20. *Caseta de dialog Settings for this Mapfile.*

În acest punct, salvați harta imagine pe care ați desenat-o. Apăsați butonul Save de pe bara cu instrumente sau alegeți File | Save din meniu ca să deschideți caseta de dialog Settings for this Mapfile, arătată în Figura 14.20. Completați adresa URL prestabilită (pentru toate părțile hărții imagine în afara ariilor definite), selectați ce tip de hartă imagine doriți să creați (CERN, NCSA sau o hartă imagine ce ține de client [CSIM]) și apăsați apoi butonul OK.

Aceasta deschide caseta de dialog Save the Image Map File. În Figura 14.21, sunt selectate setările HTML deoarece am ales CSIM în caseta de dialog anterioară. Puteți încă alege unul dintre celelalte tipuri de fișiere (CERN sau NCSA). Alegeți care vă convine cel mai mult și apăsați butonul Save.

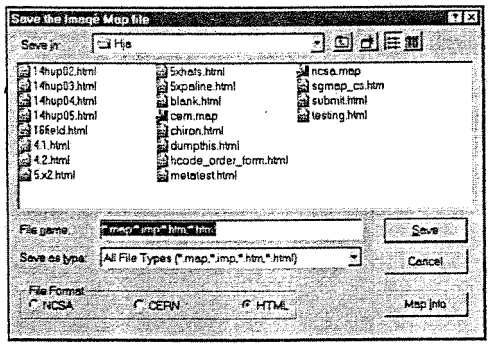


FIGURA 14.21. *Caseta de dialog Save the Image Map File.*

Iată o listă cu instrumentele pentru crearea hărților disponibile pe Web:

Coffee Cup Image Mapper++

`http://www.coffeecup.com/mapper/`

Live Image

`http://www.mediatec.com/`

Map Edit

`http://www.boutell.com/mapedit/`

Map This!

`http://www.download.com/PC/Result/TitleDetail/0,4,0-16965,501000.html`

MapMaker32

`http://members.tripod.com/~mharing/mapmaker.html`

Rezumat

Hărțile imagine sunt cea mai răspândită metodă pentru crearea meniurilor pe World Wide Web. În esență, ele sunt versiuni extinse ale hiperlegăturilor standard bazate pe imagini. Cele mai bune imagini pentru a fi folosite în hărțile imagine sunt acelea care conțin o reprezentare grafică a informațiilor existente. Hărțile imagine pot ține de client sau de server, dar tehnica actuală a celor ce țin de client prezintă avantaje imense. Totuși, amândouă tipurile pot fi amestecate într-un singur sit Web.

Hărțile imagine folosesc trei tipuri de forme: dreptunghiuri, cercuri și poligoane. Poligoanele sunt orice formă cu mai multe laturi în afară de dreptunghi. Hărțile imagine sunt, de obicei, realizate cu elementul `IMG`, dar acum a apărut o tehnică nouă, bazată pe elementul `OBJECT`. Complexitatea setării parametrilor pentru formele hărții imagine poate fi înlăturată prin folosirea unuia dintre diferitele instrumente software disponibile pentru crearea hărților imagine.

Capitolul 15
Crearea și utilizarea formularelor HTML

În acest capitol

- Atributele elementului FORM 195
- Elementul INPUT 196
- Elementul BUTTON 204
- Elementele SELECT și OPTION 205
- Elementul TEXTAREA 208
- Elementul LABEL 208
- Elementele FIELDSET și LEGEND 209
- Navigarea cu tab 211
- Taste de acces 211

ATRIBUTE COMUNE

Majoritatea elementelor din acest capitol au atribute HTML comune, cum ar fi articolele care controlează identificarea, stilul, limbajul și tratarea evenimentelor. Deoarece atributele comune sunt atât de des întâlnite în HTML 4 (de aceea sunt numite „comune"), veți găsi o prezentare a lor în Introducere, la începutul acestei cărți. Puteți de asemenea găsi mai multe informații despre aceste caracteristici des întâlnite și elementele lor asociate în Anexa A.

Între timp, în acest capitol, mă voi referi la aceste atribute comune folosind următoarele cuvinte cheie: core (principale), international (internaționale) și event (evenimente).

În plus, multe elemente pot accepta alte elemente în cadrul granițelor lor. Acestora li se va spune *elemente inline* și includ următoarele opțiuni:

Elemente `font`: Aceste elemente includ etichetele pentru bold (<B>) și italic (<I>) și sunt discutate în Capitolul 17, „Stil simplu cu text".

Elemente `phrase`: Aceste elemente includ etichete cum ar fi cele pentru evidențiere (<EM>), cod (<CODE>) și adresă (<ADDRESS>) și sunt discutate în Capitolul 7.

Elemente `special`: Această colecție este o amestecătură de elemente care include următoarele: miniaplicații și obiecte (Capitolul 16), imagini (Capitolul 11), ancore (Capitolul 13), hărți imagine (Capitolul 14), scripturi (Capitolul 23 și 24), fonturi, superscripturi și subscripturi (Capitolul 17), rânduri de pauză și citate scurte (Capitolul 7) și cadre inline (Capitolul 20).

Elementele `form`: Acest set de elemente este folosit pentru a adăuga elemente unui formular, inclusiv butoane, câmpuri de text și casete de validare.

Formularele, care permit vizitatorilor sitului dumneavoastră să introducă date, sunt folosite într-o varietate de scopuri. De exemplu, pot fi folosite la procesarea comenzilor pe un sit de comerț cu amănuntul sau pot fi construite pentru a obține părerile clienților prin e-mail.

Elementul FORM conține alte câteva elemente, numite „controale", care au o varietate de metode pentru a aduna informații. Când un formular este completat și prezentat, informațiile din controalele lui active sunt transferate unui program care acționează în vederea atingerii scopului pentru care a fost creat formularul. Fiecare element din formular are atât

un nume, cât și o valoare, astfel încât datele transferate pentru procesare sunt sub forma unor perechi nume/valoare.

Deseori, procesarea datelor se face cu un program CGI. Aceste programe sunt scrise de obicei în Perl, dar pot fi scrise în orice limbaj care poate funcționa pe un server, cum ar fi C. (Vezi Capitolul 25, „Să înțelegem interfața CGI", pentru mai multe informații.)

Altă posibilitate de manevrare a datelor formularelor este transmiterea informațiilor unui program JavaScript sau VBScript, care are grijă de procesare pe sistemul clientului, eliberând astfel resursele serverului. (Vezi Capitolele 23 și 24 pentru mai multe informații.)

Rezultatele unui formular pot fi de asemenea trimise prin e-mail către webmaster (sau alt destinatar specificat) prin folosirea acțiunii `mailto` în definiția formularului.

Atributele elementului FORM

```
<FORM [action=URL][method=get|post][enctype=MIMEType][onsubmit=script]
➥ [onreset=script][accept-charset=charset][core][international][events]>
➥ FormElements</FORM>
```

Elementul FORM are două atribute principale: `action` și `method`. Atributul `action` interpretează adresa URL ca fiind un argument al său. De obicei, este vorba de o adresă URL HTTP și furnizează adresa programului folosit pentru procesarea datelor, ca în exemplul următor:

```
<Form action="http://www.sellstuff.com/cgi-bin/getmoney.pl" method="post">
...
</FORM>
```

Deși folosirea unei adrese URL HTTP este abordarea normală, nu este ceva neobișnuit folosirea unei adrese URL mailto pentru `action`, după cum se poate vedea în codul următor:

```
<FORM action="mailto:webmaster@somewhere.com" method="post">
...
</FORM>
```

ATENȚIE!

Internet Explorer se blochează când întâlnește o adresă URL mailto.

Atributul `method` poate avea valorile GET sau POST. GET este în mod oficial nerecomandată, iar POST este acțiunea favorită. GET trimite perechile nume/valoare adresei URL specificate în atributul `method`, ca un adaos al adresei URL propriu-zise, dar POST le trimite ca o secțiune diferită care urmează antetului HTTP; această secțiune diferită se numește „corp entitate".

Atributul `enctype` este folosit pentru a anunța tipul de codare al conținutului formularului. Nu este specificat de obicei, deoarece formularele folosesc tipul de conținut `application/x-www-form-urlencoded` în mod prestabilit. Totuși, există un avantaj al specificării sale când este folosită adresa URL mailto pentru atributul `action`. În cazul adresei mailto, rezultatele unui formular sunt trimise prin e-mail la adresa specificată. Dacă datele formularului sunt trimise în codarea URL, ajung în cutia dumneavoastră poștală într-o formă aproape indescifrabilă, ca în exemplul următor:

```
firstname=John&lastname=Smith&jobdesc= This%20job%20isn%27t%20fun.
```

Cu greu puteți citi acest amestec pentru a putea traduce caracterele codate.

Mai există opțiunea procesării formularului cu un editor de text, dar trebuie să dați dovadă de o răbdare infinită pentru a aplica funcția caută-și-înlocuiește. Dacă nu ați memorat toate codurile caracterelor și țineți la sănătatea dumneavoastră mintală, puteți folosi un utilitar cum ar fi DeFORM pentru a traduce mesajul într-unul comprehensibil.

Puteți, de asemenea, trimite datele prin poștă direct sub forma textului simplu, caz în care atributul enctype își ocupă locul ce i se cuvine. Folosiți-l după cum urmează:

```
<FORM action="mailto:webmaster@somewhere.com" method="post"
➥ enctype="multipart/form-data">
...
</FORM>
```

Două atribute sunt concepute special pentru lucrul cu limbajele de scriptare: `onsubmit` și `onreset`. Acestea sunt atribute de tip eveniment specifice elementului `FORM`. Evenimentul `onsubmit` apare când formularul este transmis (de obicei prin apăsarea butonului Submit). Evenimentul `onreset` este declanșat de resetarea formularului (de obicei prin activarea butonului Reset). Pentru detalii despre folosirea limbajelor de scriptare în codul HTML, vezi Partea a V-a, „Limbaje de programare pentru HTML".

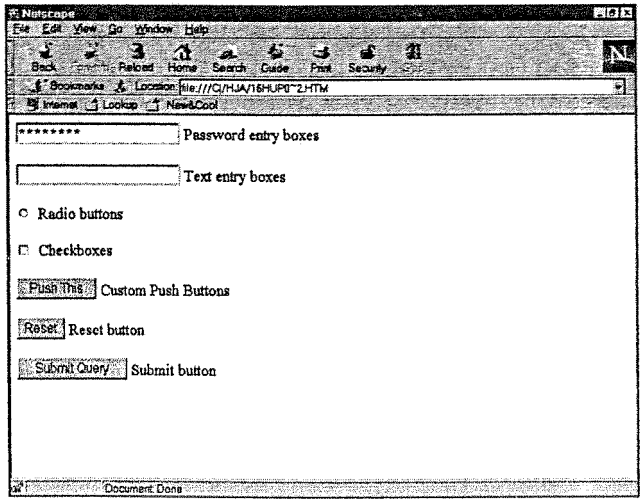
În sfârșit, noul atribut `accept-charset` este folosit pentru a specifica setul de caractere pe care serverul trebuie să le accepte pentru tratarea formularului, cum ar fi ISO-8859-9.

Elementul `INPUT`

```
<INPUT [type=text|password|checkbox|radio|submit|image|reset|button|hidden
➥|file][name=controlName][value=controlValue][checked][disabled][readonly]
➥[size=controlWidth][maxlength=wordLength][src=URL][alt=altText][usemap=URL]
➥[align=left|center|right|justify][tabindex=tabNum][accesskey=keyCombo]
➥[onfocus=script][onblur=script][onselect=script][onchange=script]
➥[accept=charset] [core][international][events]>
```

Elementul `INPUT` este cel mai important în folosirea formularelor. Este absolut posibilă construirea unui formular fără nici un alt element (vezi Figura 15.1) deoarece acesta include o gamă foarte largă de controale, care sunt specificate prin folosirea atributului `type`. Valorile acceptabile pentru atributul `type` sunt listate în Tabelul 15.1.

Figura 15.1. *Un formular construit doar cu elementul* `INPUT`.



Tabelul 15.1. Valorile atributului `type` al elementului `INPUT`.

Controalele formularului	*Valorile atributului* `type`
Butoane personale	`button`
Casete de validare	`checkbox`
Fișiere incluse	`file`
Elemente ascunse	`hidden`
Imagini	`image`
Casete de introducere a parolei	`password`
Butoane radio	`radio`
Buton Reset	`reset`
Buton Submit	`submit`
Casete de introducere a textului	`text`

Valoarea `button`

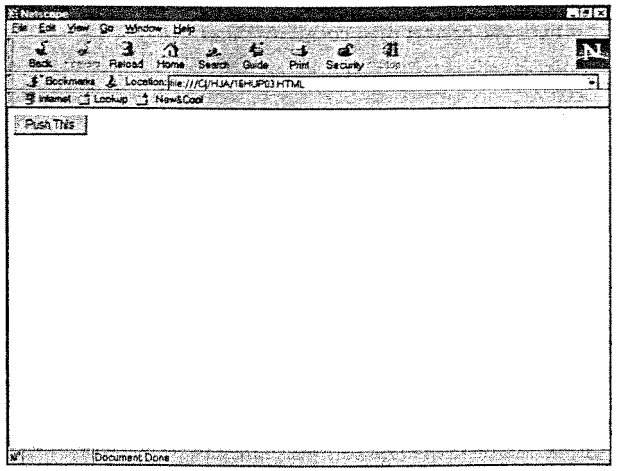
Valoarea `button` pentru atributul `type` este o noutate. Înainte de HTML 4.0, singurele butoane erau `Submit` și `Reset`, ale căror scopuri și acțiuni erau predeterminate. Intrarea tip `button` nu are funcții prestabilite; funcția pe care o îndeplinește este definită de dumneavoastră printr-un script. O declarație tipică a unui buton arată în felul următor:

```
<INPUT type="button" name="custom_button_01" value="Push This"
onclick="runprocess()">
```

Fiecărui control i se atribuie un identificator unic prin atributul `name`; fără acest identificator ar fi imposibil de spus care valoare a fost atribuită cărui control. Domeniul de existență al atributului `name` este limitat la cadrul elementului `FORM` în care se găsește.

După cum puteți vedea în Figura 15.2, textul atributului `value` este afișat pe buton. Valoarea atributului `onclick` este numele scriptului activat când este apăsat butonul.

Figura 15.2. *Valoarea* `button` *a atributului* `type`.

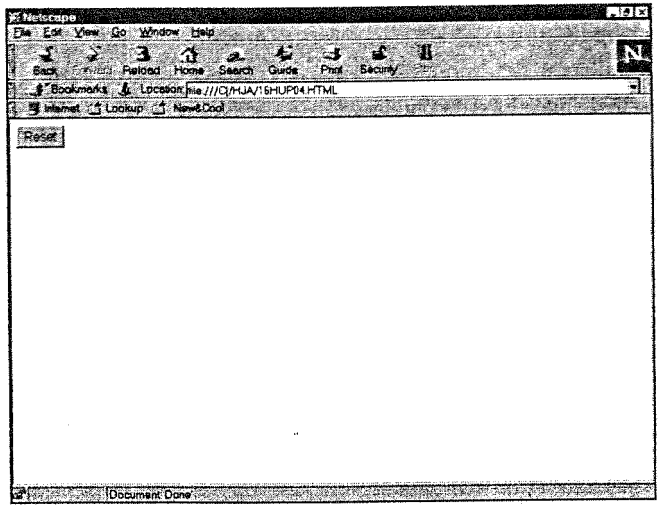


Valoarea reset

Valoarea reset a atributului type creează un buton care are un singur scop: când este apăsat, curăță toate intrările din formular, lăsându-l gol și readucând intrările la valorile lor prestabilite. (Vezi Figura 15.3.) Este declarat prin codul următor:

```
<INPUT type="reset">
```

FIGURA 15.3.
Butonul Reset.



Valoarea submit

Valoarea submit a atributului type creează un buton, arătat în Figura 15.4, care, asemănător butonului Reset, are un singur scop. În acest caz, butonul Submit trimite perechile nume/valoare ale elementelor active din formular la adresa URL specificată în declarația FORM. Destul de ciudat, specificația HTML permite existența mai multor butoane Submit. Butonul Submit este declarat prin codul următor:

```
<INPUT type="submit">
```

Valoarea image

Valoarea image a atributului type prezintă o singură diferență față de submit: înlocuiește nesăratul buton Submit cu o imagine. (Vezi Figura 15.5.) În rest, îndeplinește aceeași funcție. Codul pentru a-l declara este asemănător cu următorul:

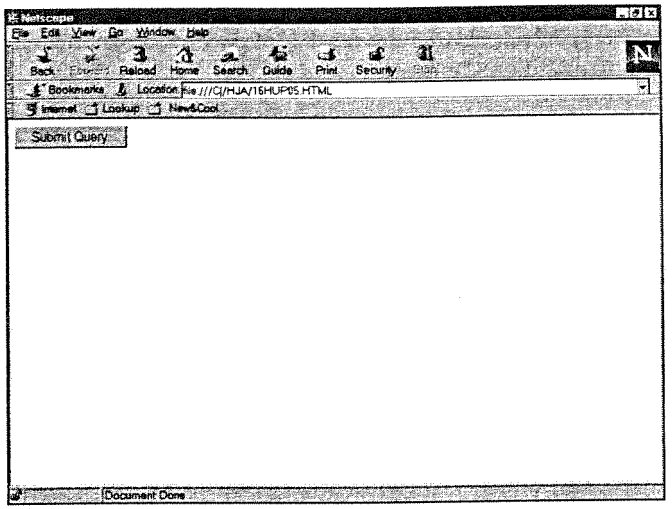
```
<INPUT type="image" src="submit_image.gif">
```

Valoarea checkbox

Valoarea checkbox a atributului type este de tip booleean; ea poate fi activă sau inactivă. Seamănă cu o casetă goală, care, o dată selectată, este umplută cu un semn de bifare pentru a arăta starea sa activă. (Vezi Figura 15.6.) Casetele de validare sunt foarte maleabile și pot fi folosite în mai multe moduri diferite. Codul de bază pentru declararea unei casete de validare este următorul:

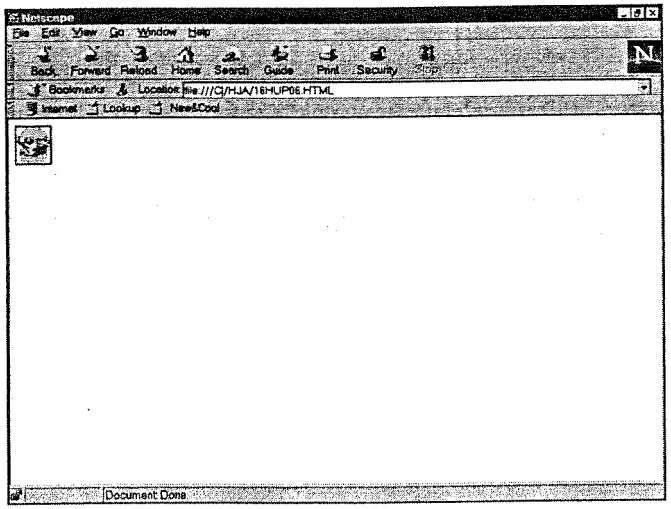
```
<INPUT type="checkbox" name="choice1" value="cards" checked>
```


FIGURA 15.4.
Butonul Submit.



BIBL. CENTR. UNIV. „M. EMINESCU" IAȘI INFORMATICĂ

FIGURA 15.5.
Un buton imagine.



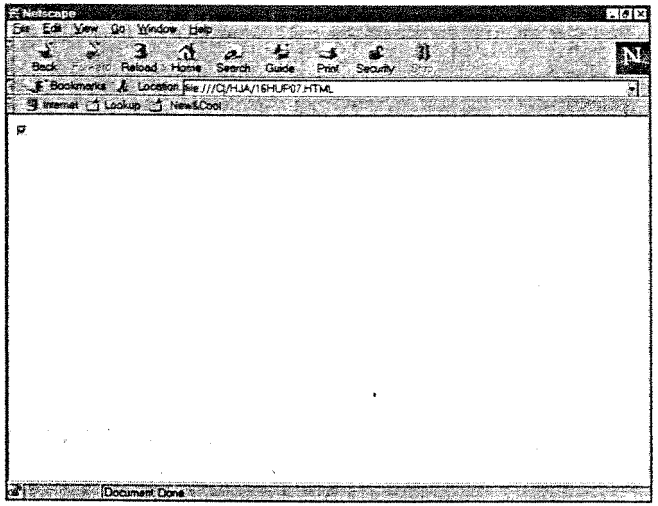
Atributul checked este opțional; el determină starea „activă" sau bifată a casetei de validare când formularul este creat. Desigur, dacă un utilizator execută un clic pe caseta de validare bifată, aceasta trece în starea sa inactivă. Atributul checked nu stabilește o stare permanentă; el doar setează starea prestabilită a casetei.

În funcție de scopurile dumneavoastră, nici atributul value nu trebuie folosit în mod obligatoriu. Dacă folosiți doar o singură casetă de validare, atributul său name este suficient. Tot ce trebuie să știți este dacă a fost bifată caseta, deoarece nici o acțiune a utilizatorului nu poate schimba conținutul atributului value (doar dacă nu vă chinuiți peste măsură să creați un foarte ciudat JavaScript). Așadar, starea sa „pornit" sau „oprit" vă poate spune tot ce

trebuie să știți. De exemplu, pentru a afla dacă cineva dorește să fie adăugat la lista de subscrieri pentru nota dumneavoastră informativă, următorul cod ar fi suficient:

```
<INPUT type="checkbox" name="addsub">
```

FIGURA 15.6.
Valoarea `checkbox` *a atributului* `type`.



Dacă un utilizator a bifat caseta, ați afla că formularul a fost trimis. Dacă nu a făcut acest lucru, atunci dumneavoastră nu ați primi nici un rezultat de la `addsub`, deoarece rezultatele elementelor goale nu sunt incluse când este trimis formularul.

ATENȚIE!

Specificația oficială HTML afirmă că atributul `value` este necesar butoanelor radio și casetelor de validare, astfel încât modul de folosire descris în această secțiune nu este, din punct de vedere tehnic, corespunzător HTML-ului.

Dacă, pe de altă parte, doreați ca utilizatorul să poată face selecții multiple la același articol, atributul `value` devine mai important. Puteți, de exemplu, să aveți orare de tipărire diferite pentru foaia dumneavoastră informativă, în care caz, ați dori să construiți ceva în genul următor:

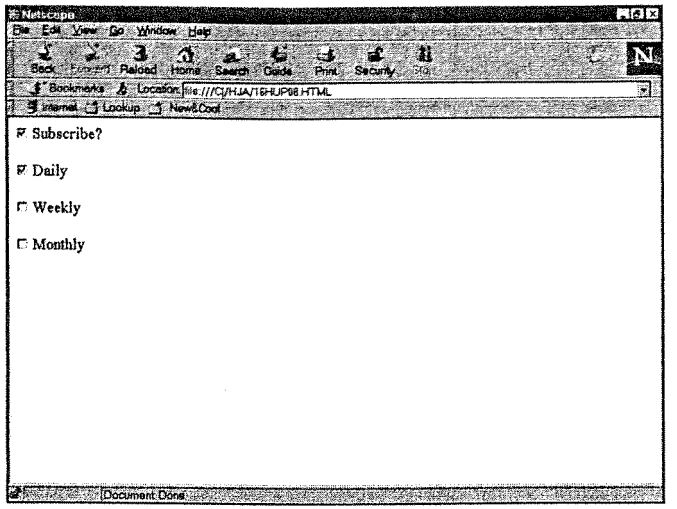
```
<INPUT type="checkbox" name="sub" value="yes" checked>
<INPUT type="checkbox" name="sub" value="daily" checked>
<INPUT type="checkbox" name="sub" value="weekly">
<INPUT type="checkbox" name="sub" value="monthly">
```

Acest cod setează formularul astfel încât comanda prestabilită ar fi o subscriere la ediția zilnică. (Vezi Figura 15.7.) Totuși, utilizatorul poate subscrie la versiunile săptămânale sau lunare – sau chiar la o combinație a celor trei, deoarece casetele nu se exclud reciproc.

Atributul `name` din acest exemplu este același pentru toate opțiunile posibile. În loc să creeze un conflict, cum s-ar întâmpla de obicei, această tehnică produce o serie de perechi nume/valoare, fiecare pereche purtând același nume, dar având valori diferite, pentru a fi trimise o dată cu expedierea formularului.

Bineînțeles, este absolut corectă și varianta în care folosiți mai multe casete de dialog având nume diferite.

FIGURA 15.7.
Mai multe casete de validare.



Valoarea `radio`

Valoarea `radio` a atributului `type` este foarte asemănătoare cu tipul `checkbox`. Diferența între cele două este că butoanele radio se exclud reciproc. Așadar, dacă ar trebui să redați opțiunile din secțiunea cu casete de validare prin butoane radio, utilizatorul nu ar putea alege decât una dintre opțiunile de subscriere: zilnic, săptămânal sau lunar, după cum se poate vedea:

```
<INPUT type="radio" name="sub" value="yes" checked>
<INPUT type="radio" name="version" value="daily" checked>
<INPUT type="radio" name="version" value="weekly" >
<INPUT type="radio" name="version" value="monthly">
```

Cu siguranță ați observat că numele din elementul `INPUT` sunt diferite. La fel ca în exemplul cu casetele de validare, utilizatorul trebuie să opteze pentru subscriere pentru a primi materialul – de unde și numele `sub`. În schimb, numele `version` este limitat la acele butoane radio care definesc versiunea pe care o dorește utilizatorul. Deoarece butoanele radio se exclud reciproc, cel selectat le deselectează automat pe celelalte, astfel încât numai selecția finală este trimisă o dată cu celelalte date ale formularului când acesta este expediat.

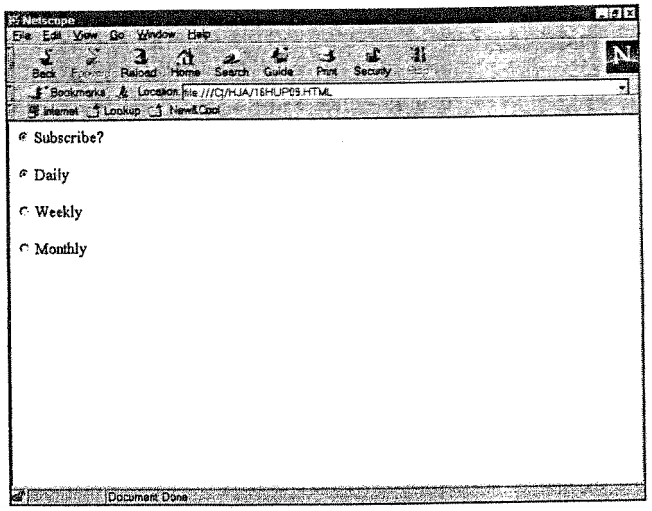
Bineînțeles, butoanele radio nu seamănă la aspect cu casetele de validare. În locul unor casete goale sau bifate, utilizatorul vede un cerc gol sau unul cu un punct mare în el. (Vezi Figura 15.8.)

Valoarea `file`

Valoarea `file` a atributului `type` nu a beneficiat niciodată de un suport prea răspândit. Specificația oficială a atribuit cinci propoziții descrierii ei. Prima oară menționată în RFC 1867 în anul 1995, este o metodă pentru adăugarea conținutului unor fișiere descărcate ale utilizatorului la ieșirea formularului. Teoretic, utilizatorului i se cere numele unui fișier. Când formularul este trimis, fișierul specificat este accesat și conținutul său este adăugat răspunsului într-un formular compus din mai multe părți.

RFC 1867 sugerează următoarea apelare:

FIGURA 15.8.
Butoane radio.




```
<FORM enctype="multipart/form-date" action="_URL_" method="post">
File to process: <INPUT name="userfile1" type="file">
<INPUT type="submit" value="Send File">
</FORM>
```

Atributul `file` ar trebui folosit cu atributul `accept`, care limitează tipurile de fișiere cărora li se permite încărcarea. Atributul `accept` primește ca argument o listă de tipuri MIME delimitate de virgule.

Valoarea `hidden`

Valoarea `hidden` a atributului `type` permite trimiterea informațiilor din formular care sunt invizibile utilizatorului. Codul de bază pentru apelarea sa este:

```
<INPUT type="hidden" name="cantsee" value="from_my_form">
```

Acest cod determină trimiterea perechilor nume/valoare `cantsee/from_my_form` la un loc cu restul informațiilor formularului.

Valoarea `text`

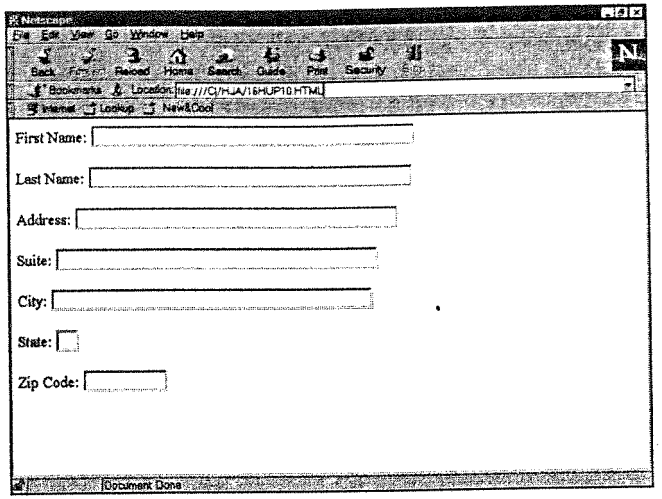
Valoarea `text` a atributului `type` cere introducerea în formular a unui răspuns pe un singur rând de lungime dată. Codul pentru adăugarea unui răspuns de tip text cu prenumele unui utilizator este următorul:

```
<INPUT type="text" name="firstname">
```

Pot fi înșiruite mai multe răspunsuri de tip text pentru a crea un formular cu o anumită semnificație, după cum se vede în următorul cod ilustrat în Figura 15.9:

```
<INPUT type="text" name="firstname" size="40">
<INPUT type="text" name="lastname" size="40">
<INPUT type="text" name="address1" size="40">
<INPUT type="text" name="address2" size="40">
<INPUT type="text" name="city" size="40">
<INPUT type="text" name="state" size="2">
<INPUT type="text" name="zipcode" size="10">
```


FIGURA 15.9.
Casete de text.



Atributul `size` fixează lungimea textului din casete în caractere. De asemenea, puteți seta numărul maxim de caractere care pot fi introduse, prin folosirea atributului `maxlength`. Dacă `maxlength` este mai mare decât `size`, atunci agentul clientului trebuie să defileze intrarea. Valoarea prestabilită a lui `maxlength` este nelimitată. Codul pentru introducerea lui `maxlength` arată cam așa:

```
<INPUT type="text" name="lastname" size="20" maxlength="40">
```

Folosind atributul `readonly`, textul poate fi declarat astfel încât utilizatorul să nu îl poată modifica. Valoarea acestui tip de text este inclusă la un loc cu valorile altor elemente active când formularul este trimis. Elementele readonly (doar pentru citire), deși nu pot fi schimbate de utilizator, sunt capabile să primească focalizarea și sunt incluse în navigarea tabulară. Acest atribut pare să aibă puține întrebuințări, deoarece atributul `hidden` deja permite autorilor de cod HTML să includă perechi nume/valoare imposibil de schimbat, având în plus avantajul de a putea fi văzute chiar și de utilizator. Este posibil, prin scriptare, să modificați dinamic starea `readonly` a unui element text, făcând astfel posibilă modificarea lui.

Valoarea `password`

Valoarea `password` a atributului `type` este identică cu atributul `text`, cu excepția că răspunsul vizibil din formular este sub forma unor asteriscuri, astfel încât nimeni nu poate citi parola secretă în timp ce se uită peste umărul utilizatorului. (Vezi Figura 15.10.) Iată un exemplu de cod pentru setarea valorii `password`:

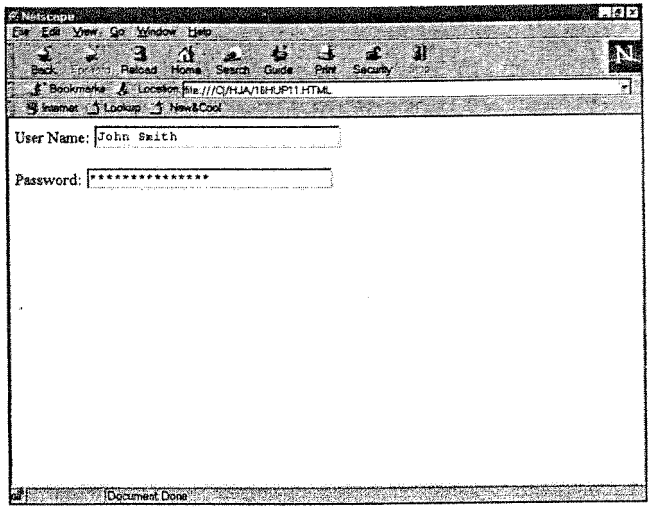
```
<INPUT type="password" name="secret" size="30">
```

La fel ca în cazul valorii `text`, puteți fixa atributele `maxsize` și `readonly`.

Alte atribute

În strânsă legătură cu `readonly` este atributul `disabled` (dezactivat). Când un element este dezactivat, conținutul său nu este numai imposibil de schimbat, dar și imposibil de folosit, astfe încât este scos din ordinea de tabulare. Când un formular este trimis, perechile nume/valoare ale elementelor de formular dezactivate nu sunt incluse. Dacă elementul dezactivat este un buton (fie el Custom, Reset sau Submit), atunci butonul nu poate fi activat. Starea de dezactivare poate fi schimbată prin scriptare.

FIGURA 15.10.
Crearea unui câmp de parolă.



În timp ce `readonly` este limitat la tipurile `text` și `password` în cadrul elementului `INPUT`, disabled poate fi aplicat oricărui tip de `INPUT`, ca și elementelor `TEXTAREA`, `SELECT`, `OPTION`, `OBJECT`, `LABEL` și `BUTTON`.

Atributul alt este folosit pentru acele câteva browsere Web care nu pot accepta formulare. Are aceeași funcție ca și atributul alt din imagini: afișează o explicare a elementului pentru utilizatorii care au browsere atât de limitate. Utilitatea acestui atribut rămâne un mister, deoarece el doar îi lasă pe oameni să vadă ce elemente conține formularul, dar nu le permite să interacționeze cu acele elemente în nici un fel. Codul pentru setarea sa arată în felul următor:

```
<INPUT alt="This is a submit button." type="submit">
```

Atributul `usemap` poate fi de asemenea folosit pentru a include hărți imagine ce țin de client în elementul `INPUT` (vezi Capitolul 14, „Crearea meniurilor cu hărți imagine"). Această întrebuințare nu a fost niciodată prea răspândită și are o utilitate limitată într-un formular. Sintaxa este următoarea:

```
<INPUT usemap="#mapx">
```

OBSERVAȚIE

Elementul `ISINDEX` acum este nerecomandat în favoarea elementului `INPUT`. El a fost folosit în timpul căutării interactive a unei pagini Web. În general, era plasat într-un document construit de un script CGI special și cerea utilizatorului să introducă un șir de căutare. Textul furnizat de utilizator era adăugat la adresa URL a documentului și trimis înapoi scriptului CGI pentru procesare sub forma `URL?search1+search2+...+searchN`. Elementul `ISINDEX` are un singur atribut, `prompt`, care schimbă mesajul prestabilit pentru câmpul de text. Sintaxa este `<ISINDEX prompt="string">`.

Elementul BUTTON

```
<BUTTON [name=controlName][value=controlValue] [type=button|submit|reset|
➥[disabled][tabindex=tabNum][accesskey=keyCombo]
➥[onfocus=script]
➥[onblur=script] [core][international][events]>ButtonText</BUTTON>
```


În elementul `BUTTON` atributul `type` poate avea trei valori posibile: `submit`, `reset` și `button`. Aceste trei valori sunt duplicate; perechile lor există deja în elementul `INPUT`, cu aceleași tipuri de nume. Ele au același efect ca și valorile corespondente din elementul `INPUT`: `submit` cauzează trimiterea formularului, `reset` resetează formularul și `button` creează un buton de apăsare definit prin script.

Este greu de spus cărui scop ar trebui să servească această duplicare a efortului. Probabil că se află în planurile W3C, pentru o dată ulterioară, eliminarea tipurilor de butoane din elementul `INPUT`.

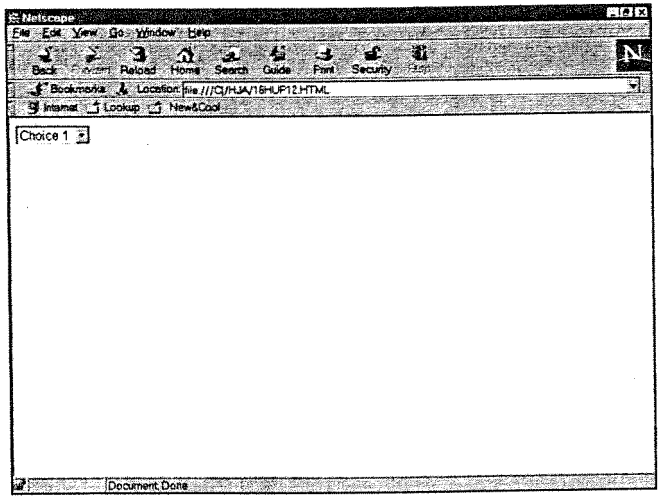
Elementele SELECT și OPTION

```
<SELECT [name=controlName][size=controlWidth][multiple][disabled]
➥[tabindex=tabNum][onfocus=script][onblur=script][onchange=script][core]
➥[international][events]></SELECT>

<OPTION [selected][disabled][value=controlValue][core][international]
➥[events]></OPTION>
```

Elementul `SELECT` este folosit pentru crearea unei liste de opțiuni, fie ca un meniu care se desfășoară, fie ca o casetă cu listă. Fiecare dintre opțiunile din listă este un element `OPTION`. Pentru a crea lista de selecții arătată în Figura 15.11, folosiți sintaxa următoare:

```
<SELECT name="testlist">
<OPTION>Choice 1</OPTION>
<OPTION>Choice 2</OPTION>
<OPTION>Choice 3</OPTION>
<OPTION>Choice 4</OPTION>
<OPTION>Choice 5</OPTION>
<OPTION>Choice 6</OPTION>
</SELECT>
```

FIGURA 15.11.
Elementul `SELECT`.



SUGESTIE

Listele de selecții sunt mai folositoare când sunt multe opțiuni de ales. Dacă acestea sunt puține, atunci casetele de validare sau butoanele radio ar putea reprezenta o alegere mai bună.

Etichetele de închidere ale elementului `OPTION` sunt opționale. Dacă nu sunt folosite, elementul se termină automat la începutul următorului element `OPTION` sau la sfârșitul elementului `SELECT` care îl conține.

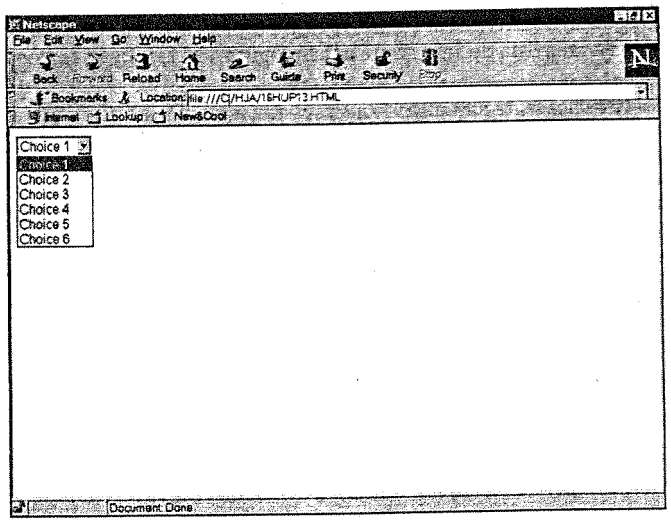
Elementul `OPTION` poate avea, de asemenea, un atribut `value` specificat, dar nu este obligatoriu. Dacă acesta este absent, conținutul elementului `OPTION` va deveni partea „value" (valoare) a perechii nume/valoare. Așadar, valoarea implicată în exemplul precedent va fi `Choice 1` pentru prima opțiune, deși nu are un anume atribut `value` atașat lui. Codul următor prezintă sintaxa pentru stabilirea atributului `value`:

```
<OPTION value="large_monitor">45 Inch Diagonal Monitor</OPTION>
```

Dacă doriți să specificați o selecție prestabilită, atunci puteți folosi atributul `selected`. Această alegere cauzează evidențierea elementului selectat. Utilizatorul o poate accepta sau o poate deselecta, după cum dorește, putând fi selectate mai multe opțiuni. Codul următor și Figura 15.12 ilustrează folosirea atributului `selected`:

```
<OPTION selected>Choice 1</OPTION>
```

FIGURA 15.12.
Atributul `selected`.

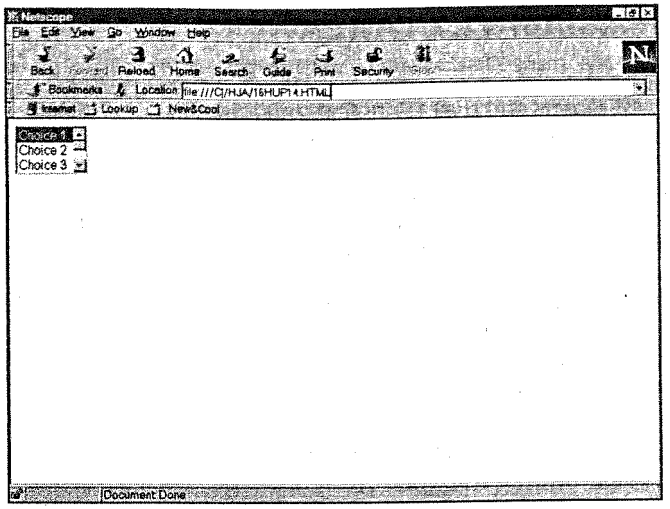


În mod prestabilit, elementul `SELECT` afișează un meniu autoderulant în care numai primul element este afișat. Totuși, puteți folosi atributul `size` pentru a face mai multe opțiuni vizibile. Prin adăugarea acestui atribut primului rând din exemplul anterior, după cum se arată în codul următor, rezultă lista de selecție prezentată în Figura 15.13:

```
<SELECT name="testlist" size="3">
```

Utilizatorii pot selecta doar o singură opțiune din meniul desfășurat, dar puteți folosi atributul `multiple` pentru a le permite selectarea unui domeniu de opțiuni. Desigur, ei pot alege în continuare numai una singură dacă așa doresc, dar, în același timp, pot alege toate opțiunile disponibile. Dacă sunt alese mai multe opțiuni, atunci sunt trimise perechi multiple nume/valoare când este expediat și formularul. Fiecare are același nume, dar au valori diferite.

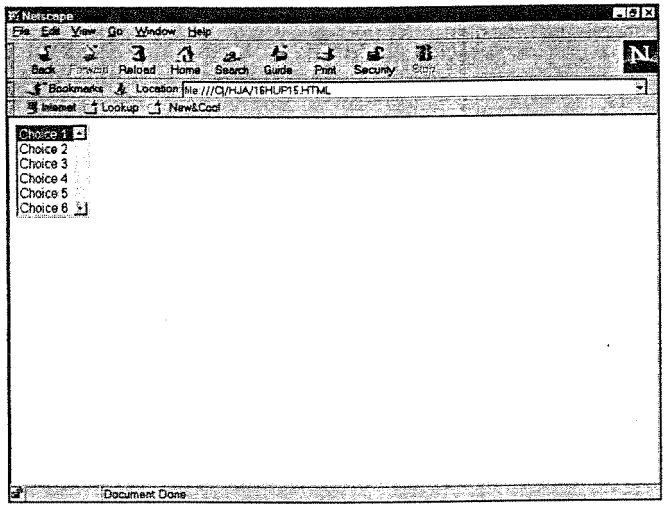
FIGURA 15.13.
Atributul `size`.



Folosirea atributului `multiple` schimbă formatul listei de selecții dintr-un meniu care se desfășoară într-o casetă cu listă. Figura 15.14 ilustrează același exemplu după întrebuințarea următorului rând:

```
<SELECT name="testlist" multiple>
```

FIGURA 15.14.
Atributul `multiple`.



OBSERVAȚIE

Nu puteți opține o casetă cu listă prin setarea atributului `size` la numărul de opțiuni. Această setare generează doar un meniu autoderulant cu bară de defilare colorată cu gri. Singurul mod de a obține o casetă cu listă este folosirea elementului `multiple`.

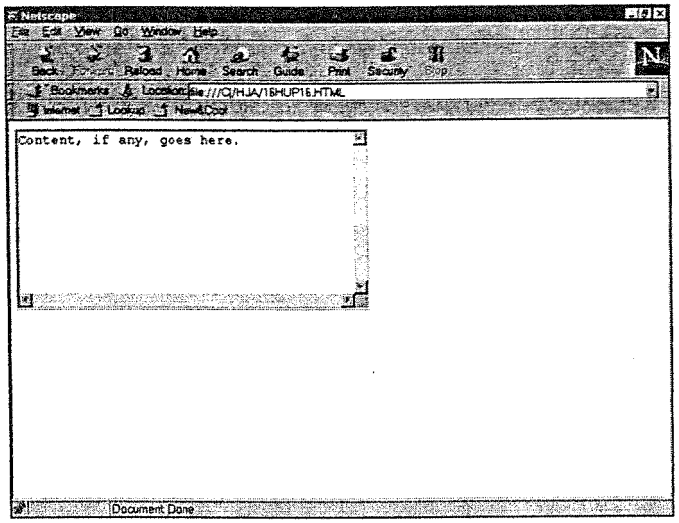
Elementul TEXTAREA

```
<TEXTAREA [name=controlName][rows=numRows][cols=numCols][disabled][readonly]
➥[tabindex=tabNum][onfocus=script][onblur=script][onselect=script]
➥[onchange=script][core][international][events]> </TEXTAREA>
```

Elementul TEXTAREA este asemănător cu tipul `text` al elementului INPUT. Diferența este că utilizatorii pot tasta într-o secțiune mai mare de text decât în cazul casetelor de text. În locul unui singur rând de text, apare o fereastră mare în care pot fi introduse răspunsuri pe mai multe rânduri. Este folosit de obicei pentru comentarii sau instrucțiuni de „livrare" – orice necesită mai mult decât un simplu răspuns. Dimensiunile ferestrei sunt specificate prin atributele `rows` și `cols`. Desigur, aceste atribute se referă la numărul de rânduri și coloane din fereastra de text. Codul următor ilustrează folosirea elementului TEXTAREA; rezultatele sunt prezentate în Figura 15.15:

```
<TEXTAREA name="comments" cols="40" rows="10">
Content, if any, goes here.
</TEXTAREA>
```

FIGURA 15.15.
Elementul TEXTAREA.



TEXTAREA necesită etichetele de început și de sfârșit. Ca în cazul tipurilor `text` și `password` ale elementului INPUT, elementele TEXTAREA folosesc atributul `readonly`. Dacă este specificat, conținutul pe care îl furnizați inițial nu va putea fi modificat de utilizator.

Elementul LABEL

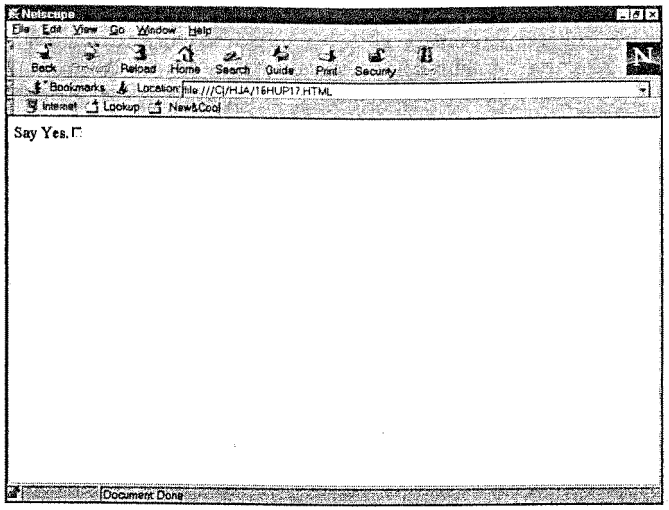
```
<LABEL [for=control-name][disabled][accesskey=keyCombo][onfocus=script]
➥[onblur=script][core][international][events]></LABEL>
```

Elementul LABEL este o noutate introdusă de HTML 4.0. După cum denotă numele, este textul care etichetează un control. Totuși, spre deosebire de textul normal, eticheta și controlul asociat ei împart aceeași focalizare. Cu alte cuvinte, executarea unui clic sau navigarea cu Tab până la etichetă are același efect ca executarea unui clic chiar pe control.

Etichetele sunt asociate controalelor printr-o metodă implicită sau explicită. În metoda de asociere implicită, elementul asociat este conținut în elementul LABEL, după cum se poate vedea în Figura 15.16 și în codul următor:

```
<LABEL>Say Yes.<INPUT type="checkbox" name="yesbox"></LABEL>
```

FIGURA 15.16.
O etichetă implicită.



În cazul asocierii explicite, eticheta este legată prin folosirea atributului `id` al elementului de control. Atributului `for` al elementului LABEL i se atribuie valoarea atributului `id` al elementului de control, după cum se arată în codul următor:

```
<LABEL for="nope">Say No.</LABEL>
<INPUT type="checkbox" name="nobox" id="nope">
```

Elementul LABEL trebuie în continuare să apară înaintea controlului pentru ca afișarea să se facă corect; asocierea logică nu are nici o legătură cu plasamentul vizual. Precedentul cod explicit produce aceeași înfățișare ca și codul implicit, al cărui rezultat l-ați văzut în Figura 15.16.

Un avantaj al abordării asocierii explicite este că elementele LABEL și cele de control sunt separate, astfel încât plasamentul lor poate fi structurat separat într-un tabel.

Specificația HTML 4.0 nu are în vedere posibilitatea unui conflict între atributele `name` și `id`.

ATENȚIE!

LABEL este foarte nou și nu pe deplin implementat. Netscape Navigator 4.03 nu are suport pentru el, iar Internet Explorer 4.0, deși are suport pentru declararea explicită a etichetelor, nu acceptă relațiile lor implicite.

Elementele FIELDSET și LEGEND

```
<FIELDSET [core][international][events]></FIELDSET>

<LEGEND [align=left|center|right|justify][accesskey=keyCombo][core]
➥[international][events]></LEGEND>
```

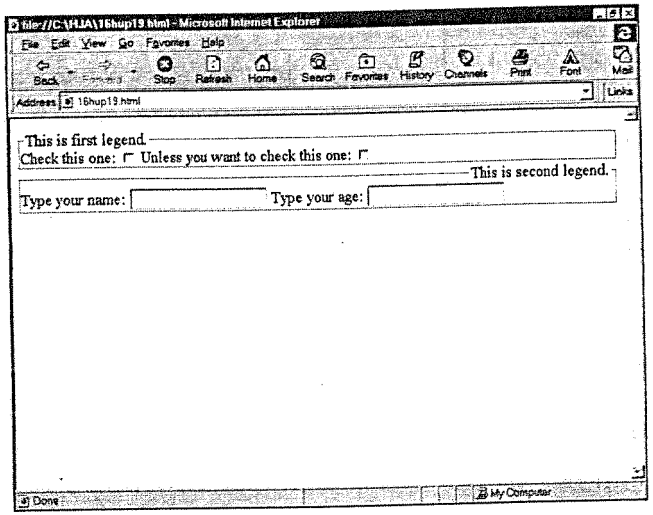
Elementele FIELDSET și LEGEND sunt și ele noutăți ale versiunii HTML 4.0. Seturile de câmpuri creează casete în jurul controalelor grupate, la fel cum fac panourile în programarea standard, iar legendele sunt etichete care se referă la totalitatea setului de câmpuri. Amândouă elementele necesită etichete de început și de sfârșit. Folosirea seturilor de câmpuri și a legendelor lor este ilustrată în Figura 15.17 și în codul următor:

```
<FIELDSET>
<LEGEND align="left">This is first legend.</LEGEND>
Check this one: <INPUT type="checkbox" name="firstbox">
Unless you want to check this one: <INPUT type="checkbox" name="otherbox">
</FIELDSET>
<FIELDSET>
<LEGEND align="right">This is second legend.</LEGEND>
Type your name: <INPUT type="text" name="typename">
Type your age: <INPUT type="text" name="typeage">
</FIELDSET>
```

Valorile posibile pentru atributul align sunt top, bottom, right și left. Alinierea în acest caz este în raport cu setul de câmpuri, nu cu pagina.

Nu există nici o modalitate în specificația HTML 4.0 pentru controlarea mărimii seturilor de câmpuri; acestea se întind pe tot ecranul, indiferent de spațiul ocupat de controalele pe care le conțin.

FIGURA 15.17. *Elementele FIELDSET și LEGEND.*



ATENȚIE!

FIELDSET și LEGEND sunt foarte noi și implementate incomplet. Netscape Navigator 4.03 nu are suport pentru ele, iar Internet Explorer 4.0 nu aliniază corect legendele. Deși right plasează legenda în aria de sus-dreapta a setului de câmpuri, top, left și bottom trimit toate legenda în aria de sus-stânga a setului de câmpuri.

Navigarea cu Tab

Atributul tabindex este pentru navigarea printr-un formular folosind tasta Tab pentru a ne muta de la un element la următorul. Nu aveți nevoie să folosiți tabindex dacă vreți ca formularul să fie parcurs în ordinea în care apar elementele, după cum este de obicei cazul. Dacă tabindex nu este specificat, atunci acesta va fi comportamentul prestabilit al tastei Tab.

Totuși, dacă doriți să navigați în altă ordine sau dacă doriți ca niște elemente să fie date la o parte din ordinea de tabulare, atunci acest atribut devine folositor. Pentru a seta ordinea de tabulare pentru o pereche de casete de introducere a textului, veți folosi următoarea sintaxă:

```
<INPUT tabindex="1" type="text" name="firstname">
<INPUT tabindex="2" type="text" name="lastname">
```

Dacă două elemente au aceași valoare a lui tabindex, atunci ordinea de navigare va fi ordinea în care apar. Codul următor ar avea atunci același efect ca exemplul precedent:

```
<INPUT tabindex="1" type="text" name="firstname">
<INPUT tabindex="1" type="text" name="lastname">
```

Nu este necesar ca valorile atribuite să fie consecutive; tot ce trebuie este ca ele să fie în ordine crescătoare, în conformitate cu ordinea de navigare pe care o doriți. Așadar, codul următor are același efect ca și cele două anterioare:

```
<INPUT tabindex="11" type="text" name="firstname">
<INPUT tabindex="32" type="text" name="lastname">
```

De fapt, ar fi inteligent din partea dumneavoastră să atribuiți valori neconsecutive atributelor tabindex ale elementelor succesive. Această tehnică a fost intens folosită în zilele de demult ale programării în BASIC, când fiecărui rând i se atribuia propriul număr, scopul fiind să se lase loc pentru introducerea unor noi rânduri de cod după cum era nevoie, fără a mai fi necesară renumerotarea tuturor. Dacă ați atribuit 1 și 2 ca valori ale lui tabindex pentru două elemente și doriți apoi să introduceți alt element între ele, v-ați dat singur de lucru suplimentar. Dacă, pe de altă parte, valorile lui tabindex sunt 100 și 200, puteți adăuga toate elementele de care ați putea avea nevoie fără să mai trebuiască să le renumerotați pe cele deja existente.

Pentru a înlătura un element din ordinea de navigare tabulară, puteți folosi atributul disable. Deși, dacă doriți să îl înlăturați complet din ordine, dar fără să îl dezactivați, puteți să îi atribuiți lui tabindex o valoare negativă, ca în exemplul următor:

```
<INPUT tabindex="-1" type="text" name="firstname">
```

Atributul tabindex este acceptat de elementele BUTTON, INPUT, SELECT și TEXTAREA (este de asemenea acceptat și de elementele ANCHOR, AREA și OBJECT, dar acestea nu sunt elemente de formular).

Taste de acces

Tastele de acces sunt discutate pe larg în Capitolul 41. Ele sunt folosite pentru a furniza acces direct de la tastatură, în acest caz, la elementele de formular. Unui element i se atribuie o anume tastă fierbinte, iar atunci când este apăsată tasta respectivă împreună cu tasta Alt (pentru utilizatorii de IBM) sau Cmd (pentru utilizatorii de Apple), elementele FORM sunt activate. Utilizatorii sunt deja familiarizați cu această tehnică, folosită în general pentru selectarea meniurilor, precum și în multe alte scopuri.

Pentru a stabili o tastă de acces la un element de formular, folosiți atributul accesskey; ca valoare a sa, specificați caracterul tastei fierbinți pe care doriți să o atribuiți elementului. Iată un exemplu de stabilire a unei taste de acces:

```
<INPUT type="text" name="address" accesskey="A">
```

Trebuie să țineți o evidență strictă a tastelor de acces pe care le-ați atribuit, pentru a nu crea dubluri care cauzează conflicte între elemente.

ATENȚIE!

Tastele de acces sunt foarte noi și incomplet implementate. Netscape Navigator 4.03 nu are suport pentru ele, iar Internet Explorer 4.0 nu recunoaște câteodată tastele de acces dacă nu sunt codate ca litere capitalizate. Există, de asemenea, o greșeală fundamentală de design a tastelor de acces – ele anulează tastele de acces ale browserului. Așadar, dacă codați o tastă de acces cu F, nu veți putea folosi tastatura pentru a selecta meniul File al browserului.

Rezumat

Formularele vă permit să primiți ușor intrări de la vizitatorii sitului dumneavoastră. Întrebuințările formularelor sunt, din punct de vedere practic, limitate doar de creativitatea cu care le aplicați dumneavoastră. Elementul general FORM este folosit pentru a găzdui alte câteva elemente, numite „controale". Informațiile din controale (sub forma unor perechi nume/valoare) sunt procesate printr-un program separat când este trimis formularul. Destul de des, această procesare este realizată de un program CGI, dar acest lucru se poate face și cu un limbaj de scriptare, cum ar fi JavaScript. Datele pot fi, de asemenea, trimise prin e-mail la o adresă predeterminată.

Controalele au o gamă mare de capacități. Elementul INPUT, de exemplu, poate fi codat sub forma unor casete de text, butoane, casete de dialog, butoane radio și altele. Butoanele Reset și Submit (care curăță formularul, respectiv trimit conținutul acestuia pentru procesare) sunt și ele exemple de întrebuințări ale elementului INPUT. Cu elementele SELECT și OPTION, puteți construi o listă de selecții, sub forma unui meniu care se desfășoară sau a unei casete cu o listă, pentru a putea alege diferite opțiuni. Elementul TEXTAREA vă permite să rezervați o arie din formular în care utilizatorii să poată introduce bucăți mari de text; de obicei este folosit pentru secțiunea de comentarii. Elementul LABEL creează o etichetă de tip text care, spre deosebire de textul normal, are aceeași focalizare ca și controlul căruia îi este asociată. Elementele FIELDSET și LEGEND separă grafic secțiunile formularului în același fel în care panourile fac acest lucru în programe. Setul de câmpuri delimitează un grup de elemente în jurul căruia este trasă o linie, iar legenda dă o etichetă întregului set de câmpuri.

Formularele au prestabilită navigarea tabulară de sus în jos, dar dumneavoastră puteți schimba ordinea în care tabularea va focaliza elementele, dacă doriți. Puteți, de asemenea, să atribuiți taste de acces la elementele din formular, permițând utilizatorilor să focalizeze un element prin apăsarea unei anume combinații de taste.

Capitolul 16
Obiecte înglobate în HTML

de Rick Darnell

În acest capitol

- **Adăugarea unui conținut specializat pe o pagină Web 213**
- **Întrebuințarea de bază a etichetei <OBJECT> 214**
- **Adăugarea unui conținut dinamic cu <OBJECT> 215**
- **Folosirea altor obiecte și a caracteristicilor speciale 219**

Dacă doriți să prezentați și altceva în afara textului și a liniilor orizontale în HTML, va trebui să le inserați pe pagină din alt fișier. Aceasta este o realitate în HTML care nu are de gând să se schimbe pentru mult, mult timp.

HTML 3.2 este limitat la două etichete pentru inserarea conținutului special, deosebit de text: eticheta <APPLET>, pentru adăugarea miniaplicațiilor Java, și eticheta <IMG>, pentru inserarea imaginilor grafice. HTML 4.0 a evoluat dincolo de cele două etichete originale, prin adăugarea etichetei Microsoft <OBJECT> și modificarea ei pentru a fi folosită de toate conținuturile speciale, inclusiv imagini, miniaplicații, plug-in-uri și altele. Această adăugire înseamnă de asemenea că eticheta Netscape <EMBED> nu va căpăta niciodată recunoaștere oficială din partea Consorțiului World Wide Web (W3C).

Acest capitol explorează metode de întrebuințare a etichetei <OBJECT> pentru inserarea celor mai populare tipuri de conținut disponibile acum pe Web și arată cum poate fi folosită eticheta pentru a adăuga noi tipuri de conținut în viitor, fără altă revizuire din partea W3C.

Adăugarea unui conținut specializat pe o pagină Web

O dată cu progresul browserelor și al suportului aplicațiilor, lista articolelor pe care le puteți adăuga unei pagini Web crește de la săptămână la săptămână. Conținutul include articole acceptate direct de un browser, cum ar fi GIF-uri animate, Portable Network Graphics (PNG), miniaplicații Java și realitate virtuală. De asemenea, include articole acceptate prin aplicații asistent (helper) și plug-in-uri, aplicații speciale cărora browserul le permite rularea în fereastra sa. Atât plug-in-urile Netscape, cât și componentele Microsoft ActiveX sunt exemple ale acestui tip de conținut. Cu aceste instrumente, puteți vedea fișiere ale programelor de calcul tabelar și ale procesoarelor de text sau puteți reda secvențe audio și video, de exemplu.

În funcție de capacitățile browserului, fiecare obiect de pe o pagină Web poate fi, la rândul său, tratat intern de browser sau de aplicația asistent invocată. În orice caz, conținutul rămâne ca parte a restului paginii Web.

CUM ȘTIE UN BROWSER CE SĂ FACĂ?

Când browserul dumneavoastră încarcă un anumit fișier – indiferent dacă este vorba de un fișier HTML, o imagine sau multimedia – cum știe ce să facă cu el? Reacția browserului este determinată de două lucruri: extensia fișierului și tipul de conținut al fișierului. *(continuare)*

(continuare)

Veți vedea multe extensii de fișier în exemplele din această carte, printre care `.html` (pagini Web), `.gif` (imagini), `.dcr` (Shockwave) și `.wrl` (realitate virtuală). Browserul folosește extensiile fișierelor pentru a determina ce să facă atunci când recuperează un fișier de pe discul dumneavoastră local.

Tipul conținutului este folosit când browserul primește un fișier de pe un server Web, deoarece serverul Web nu trimite întotdeauna numele fișierului. În unele cazuri, serverul trimite numai datele, fără nici o informație despre fișier. În locul unui nume de fișier și al unei extensii, trimite un cod special, numit *content type* (tipul conținutului), care identifică informațiile pe care le trimite. Tipurile de conținut includ `text/html`, `image/gif`, `application/mpe`, `application/msword` și altele.

Atât browserul, cât și serverul conțin liste de extensii ale fișierelor și de tipuri de conținut. Serverul folosește lista pentru a determina care tip de conținut să fie trimis cu un fișier dat. Lista browserului include o intrare adițională pentru aplicațiile asistent corespunzătoare tipurilor de conținut.

Întrebuințarea de bază a etichetei <OBJECT>

Eticheta `<OBJECT>` nu reprezintă nimic nou pentru Web. Microsoft a introdus această etichetă o dată cu apariția tehnologiei ActiveX ca o metodă de a introduce controale înglobate pe o pagină Web. Versiunea etichetei `<OBJECT>` din HTML 4.0 a fost extinsă să funcționeze cu toate coținuturile speciale, inclusiv miniaplicații, plug-in-uri, ActiveX, imagini și oricare alt tip de conținut înglobat.

Sintaxa de bază propusă este următoarea:

```
<OBJECT data=URL type=MIMEtype height=pixels width=pixels>
 <!--Parameters-->
 <!--Alternative content, including other objects-->
</OBJECT>
```

Aici, atributul `data` spune numele fișierului, cum ar fi `image.gif`, iar atributul `type` este tipul conținutului Internet, cum ar fi `image/gif`, urmat de înălțimea și lățimea ariei obiectului în pixeli. Este urmat de un număr de etichete parametru pentru controlul adițional al conținutului, de conținutul alternativ pentru browserele incompatibile cu obiectul și de o etichetă de închidere `</OBJECT>`.

ATENȚIE!

Inserarea imaginilor și a altor obiecte cu eticheta HTML 4.0 `<OBJECT>` rămâne în mare parte neimplementată pe majoritatea browserelor. Singura excepție o reprezintă controalele ActiveX, care sunt încă acceptate cu `<OBJECT>` pe Internet Explorer.

Folosirea etichetei pentru miniaplicații sau plug-in-uri pe Internet Explorer ar putea cauza blocarea browserului. Pe Navigator, eticheta `<OBJECT>` este ignorată complet și este folosit textul alternativ.

Vă recomand să așteptați până când Navigator și Internet Explorer sunt aduse la un nivel corespunzător înainte să încercați o implementare la scară mare a lui `<OBJECT>` pe situl dumneavoastră.

De exemplu, să spunem că doriți să includeți o secvență video cu pisica dumneavoastră pe pagina Web, iar aceasta este localizată în fișierul film QuickTime denumit `festusOfTheNorth.mov`. Puteți face acest lucru folosind eticheta `<OBJECT>` în felul următor:

```
<OBJECT data="festusOfTheNorth.mov" type="video/quicktime" alt="My cat">
</OBJECT>
```

Această metodă inserează fișierul și îi identifică tipul MIME pentru browser. Dacă browserul nu știe cum să manevreze un film QuickTime, atunci nu va afișa nimic. Dar să presupunem că sunteți un autor de pagini Web conștiincios, care dorește să furnizeze un conținut

alternativ pentru browserele care nu acceptă QuickTime, astfel încât inserați o animație Microsoft, după cum urmează:

```
<OBJECT data="festusOfTheNorth.mov" type="video/quicktime" alt="My cat">
 <OBJECT data="festusOfTheNorth.avi" type="video/msvideo" alt="My cat">
 </OBJECT>
</OBJECT>
```

Acum, dacă browserul nu acceptă un film QuickTime, va încerca animația Microsoft. Să presupunem că doriți să oferiți încă câteva opțiuni: un Portable Network Graphics, un vechi GIF întrețesut și un text pentru browserele nongrafice. Atunci, imbricați toate etichetele până când ajungeți la ultimul identificator de conținut:

```
<OBJECT data="festusOfTheNorth.mov" type="video/quicktime" alt="My cat movie">
 <OBJECT data="festusOfTheNorth.avi" type="video/msvideo" alt="My cat movie">
  <OBJECT data="festusOfTheNorth.png" type="image/png" alt="My cat image">
    <OBJECT data="festusOfTheNorth.gif" type="image/gif" alt="My cat image">
    You should see my cat, Festus of the North. He's a fat black cat with white
    ➥toes.
    </OBJECT>
  </OBJECT>
 </OBJECT>
</OBJECT>
```

SUGESTIE

Ca opțiune pentru conținutul interior prestabilit al imaginii, puteți include eticheta `<IMG>` cu atributul `alt`, pentru a garanta suportul pe browserele mai vechi care încă nu au suport pentru întrebuințări multiple ale etichetei `<OBJECT>`. În exemplul precedent, cel mai central obiect și textul ar fi înlocuite cu `<IMG src="festufOfTheNorth.gif" alt="My cat, Festus of the North.">`.

Până acum, toate bune... dar cum funcționează `<OBJECT>` cu miniaplicații, plug-in-uri și controale ActiveX? Veți vedea acest lucru în secțiunea următoare.

Adăugarea unui conținut dinamic cu <OBJECT>

În plus față de conținutul static, cum ar fi imaginile, animațiile și secvențele audio, puteți, de asemenea, folosi eticheta `<OBJECT>` pentru adăugarea unui conținut interactiv, cum ar fi miniaplicațiile, plug-in-urile și controalele înglobate, printr-un atribut special, numit `classid`.

SUGESTIE

În acest capitol, definesc „conținutul dinamic" ca orice ar putea interacționa cu utilizatorul, ceea ce limitează destul de mult domeniul aplicațiilor. Deși fișierele video și audio (MPEG, AVI, AU, WAV) par a fi dinamice, ele de obicei nu pot fi controlate sau schimbate de utilizator.

Sintaxa de bază pentru adăugarea conținutului aplicațiilor unei pagini Web este următoarea:

```
<OBJECT classid="protocol:fileName" codetype="type/subtype">
<!--Alternative content for noncompatible browsers-->
</OBJECT>
```

Atributul `classid` conține protocolul și numele de fișier sau identificatorul conținutului separate de două puncte, iar `codetype` comunică ce tip de aplicație este aceasta. Între etichetele de deschidere și de închidere, puteți include, de asemenea, etichete `<OBJECT>` adiționale, un conținut alternativ sau parametrii care trebuiesc transmiși aplicației.

Parametrii sunt incluși cu eticheta `<PARAM>`. Sintaxa de bază pentru întrebuințarea sa este următoarea:

```
<PARAM name="paramName" value="paramValue">
```

Aici, *paramName* este numele parametrului, iar *paramValue* este valoarea sa asociată. Există două atribute adiționale pentru această etichetă: `valuetype` și `type`.

- **`valuetype`:** Acest atribut definește modul în care ar trebui interpretat atributul `value`. Valorile posibile pentru acest atribut sunt `data` (prestabilită), `ref` și `object`. Valoarea prestabilită îl trimite pe `value` obiectului sub forma unui șir. Cu următorul tip de valoare, `ref`, atributul `value` este tratat ca o adresă URL a unei resurse. Ultimul tip de valoare, `object`, identifică altă etichetă `<OBJECT>` pe pagina curentă prin atributul `id` al obiectului.
- **`type`:** Acest atribut, folosit numai când tipul valorii este setat la `ref`, identifică tipul MIME al resursei identificate în atributul `value`.

Cunoscând obiectele și parametrii de bază, a venit timpul să jonglăm cu cele trei tipuri de conținut pentru care au fost concepute etichetele: miniaplicații, plug-in-uri și controale (vai de noi).

Miniaplicații Java

Puteți folosi eticheta `<OBJECT>` pentru a insera una dintre cele mai cunoscute forme de conținut înglobat de pe Web – miniaplicațiile Java. Java poate implementa sunet, animație și alte articole pentru interactivitatea cu utilizatorul, indiferent de platformă. În momentul actual, limbajul Java este acceptat de cel puțin patru browsere: HotJava de la Sun (care este scris în Java), Netscape Navigator 2.0 sau o versiune ulterioară, Microsoft Internet Explorer 3.0 sau o versiune ulterioară și versiunea beta a W3C pentru serverul Jigsaw Web.

PERICOL!

Internet Explorer 4 și Navigator 4 în continuare nu acceptă eticheta `<OBJECT>` pentru miniaplicații Java. Va trebui să folosiți eticheta `<APPLET>` în cadrul etichetei `<OBJECT>` ca un conținut opțional sau să vă abțineți de la folosirea etichetei `<OBJECT>` până când implementările browserului ajung din urmă standardul.

Aceasta este cea mai simplă sintaxă pentru includerea unei miniaplicații Java cu `<OBJECT>`:

```
<OBJECT classid="java:className.fileExtension">
</OBJECT>
```

În acest caz, *className* este numele fișierului de clasă Java, iar *fileExtension* este extensia fișierului de clasă, de obicei `class`. Această sintaxă simplă necesită ca fișierul Java să se afle în același director cu pagina Web. Pentru a ajuta browserul să decidă dacă este capabil să ruleze miniaplicația ca pe un conținut executabil, W3C recomandă și folosirea atributului `codetype`:

```
<OBJECT classid="java:className" codetype="application/octet-stream">
</OBJECT>
```

Acum, browserul va privi întâi tipul codului. Dacă nu poate rula tipuri de conținut executabil simple, va renunța să încarce miniaplicația și va trece la sarcinile următoare. Altfel, va privi protocolul din `classid`. Dacă browserul poate rula o aplicație și are suport pentru Java, atunci el va continua încărcarea clasei pentru a o executa.

Dacă fișierele miniaplicației dumneavoastră se află în locații diferite față de cea a paginii Web, atunci folosiți atributul `codebase` pentru a identifica directorul cu o adresă URL:

```
<OBJECT classid="java:className" codetype="application/octet-stream"
        codebase=URL>
</OBJECT>
```

Când browserul încearcă să încarce miniaplicația, alipește în fața clasei adresa URL pentru a găsi fișierul căutat.

Unele miniaplicații pot accepta de asemenea și alți parametri pentru a le controla comportamentul. Pentru o miniaplicație ticker, ar putea fi un mesaj care se desfășoară și o imagine de fundal. Pentru o miniaplicație hartă imagine, ar putea fi o listă de adrese URL și coordonate ale imaginii. Ați înțeles ideea. Toate aceste informații suplimentare sunt incluse cu eticheta `<PARAM>`, discutată mai devreme în acest capitol.

Nu se ține cont de literele mari și mici în numele parametrului, dar trebuie să se potrivească exact cu numele parametrului din miniaplicație. Valoarea unui parametru este cu totul altă problemă. Toți parametrii transmiși dintr-o pagină HTML unei miniaplicații sunt sub formă de șiruri, indiferent de ce sunt ei sau de cum sunt formatați pe pagină. Orice conversie în alt tip (număr întreg, booleean, dată) trebuie să aibă loc în miniaplicație.

ETICHETA `<APPLET>`

Eticheta `<APPLET>` este în continuare parte oficială din HTML, deși întrebuințarea sa este nerecomandată datorită dezvoltării etichetei `<OBJECT>`. Dacă vă simțiți nostalgic și doriți în continuare să includeți miniaplicații pe pagina dumneavoastră Web prin eticheta `<APPLET>`, urmați această sintaxă:

```
<APPLET code="appletName.class"[codebase="pathToClass"]
        width=pixelWidth height=pixelHeight [align=alignment]>
  <!-- optional parameters -->
  <!-- alternative content -->
</APPLET>
```

Rândul de cod cerut identifică numele miniaplicației și mărimea cu care va apărea pe pagină. Trebuie să includeți extensia `class` în atributul `code`, iar Java Virtual Machine asociată cu browserul ar trebui automat să presupună că prima metodă de apelare este `init`.

Dacă miniaplicația se află într-un director diferit de cel al pagini Web, atunci trebuie să furnizați browserului adresa URL către locația miniaplicației prin `codebase`. De asemenea, probabil că veți dori să setați mărimea și alinierea miniaplicației prin atributele de înfățișare.

La fel ca eticheta `<OBJECT>`, eticheta `<APPLET>` poate folosi parametri prin intermediul etichetei `<PARAM>`.

Controale ActiveX

Eticheta `<OBJECT>` a fost inițial creată pentru tehnologia ActiveX a celor de la Microsoft. Reprezintă o altă opțiune pe care o au designerii de pagini Web pentru livrarea conținutului specializat utilizatorului. Acest standard, dezvoltat de Microsoft, este o încrucișare între Java și plug-in-uri. Asemănător cu Java, este încărcat automat când este întâlnit de browser; asemănător unui plug-in, rămâne încărcat pe sistemul utilizatorului în vederea folosirii viitoare și nu are nevoie să mai fie încărcat încă o dată.

În prezent, doar Java Internet Explorer și Netscape Navigator oferă suport pentru ActiveX. Alte browsere s-ar putea să le urmeze, dar nu este nici un semnal că acest lucru se va întâmpla curând.

Controalele ActiveX oferă și o varietate de opțiuni pentru conținut, inclusiv Shockwave, foi de calcul tabelar și controale tip Windows. Pentru a include un control ActiveX, folosiți eticheta `<OBJECT>` cu sintaxa următoare:

```
<OBJECT classid="clsid:identifier" data="contentFile">
</OBJECT>
```

Identifier este un șir de 36 de caractere care identifică controlul, iar `data` identifică adresa URL a fișierului în care se găsește conținutul.

SUGESTIE

Dacă folosiți atributul `codetype` pentru controlul dumneavoastră, valoarea corespunzătoare pentru ActiveX este `application/x-oleobject`.

Atributul `codebase` are o întrebuințare puțin diferită pentru ActiveX. În loc să se ducă automat la sursă, browserul verifică sistemul utilizatorului pentru a vedea dacă nu este deja încărcat controlul. Dacă este, browserul folosește versiunea locală; dacă nu este, browserul folosește `codebase` pentru a-l încărca și instala din rețea.

Plug-in-uri Netscape

În primele zile ale lui Mosaic și ale altor browsere Web timpurii, pentru a beneficia de dezvoltări ale tehnologiei Web trebuia să așteptați o nouă versiune a browserului. Dacă era introdus un nou tip de conținut, cum ar fi secvențele video sau realitatea virtuală, trebuia să așteptați până când apărea versiunea browserului dotată cu suportul necesar. Netscape a introdus plug-in-urile, care au extins capacitățile browserelor fără să se mai aștepte o nouă ediție.

Unul dintre cele mai cunoscute plug-in-uri multimedia este Shockwave de la Macromedia, care permite autorilor de pagini să afișeze conținutul multimedia dezvoltat cu Shockwave Director. Acest conținut poate lua forma animațiilor, agendelor electronice, jocurilor sau a altor articole care necesită sincronizarea sunetului și imaginii cu interactivitatea.

SUGESTIE

O problemă majoră în ceea ce privește plug-in-urile este că acestea solicită utilizatorului să aibă un browser care să le accepte și să aibă plug-in-urile necesare instalate. Multe plug-in-uri sunt disponibile doar pentru una sau două platforme, utilizatorii celorlalte platforme rămânând cu ochii în soare. De asemenea, trebuie să vă coordonați cu administratorul de sistem pentru a vă asigura că serverul Web știe cum să livreze marfa.

Netscape include câteva dintre cele mai cunoscute plug-in-uri, cum sunt cele pentru realitatea virtuală și conversațiile pe Web, prin versiunea standard de software. Microsoft include, de asemenea, controale ActiveX în Internet Explorer pentru a detecta și implementa plug-in-urile.

Înglobarea plug-in-urilor în pagina dumneavoastră Web prin eticheta `<OBJECT>` este similară cu adăugarea imaginilor. Începeți cu numele fișierului de conținut al plug-in-ului și tipul său:

```
<OBJECT data="filename" type="contentType/contentSubtype">
</OBJECT>
```

Această sintaxă ar trebui să funcționeze pentru majoritatea conținuturilor plug-in-urilor care sunt depozitate în același director cu pagina sursă. Pentru un fișier plug-in depozitat altundeva în situl Web, puteți adăuga atributul `codebase`. Celelalte atribute de înfățișare se aplică și ele, inclusiv cele pentru mărime și aliniere.

De exemplu, un plug-in Macromedia Shockwave ar putea fi inserat în felul următor:

```
<OBJECT data="movingEye.dcr" type="application/x-director" codebase="/specialStuff"
        height=244 width=200>
  This animation requires <A href="www.macromedia.com"
  title="Macromedia">Macromedia Shockwave</A>.
</OBJECT>
```

Browserul va încerca să încarce fișierul la `/specialStuff/movingEye.dcr`, care este un tip MIME `application/x-director`. Dacă browserul nu acceptă tipul MIME, atunci este afișat conținutul alternativ, care direcționează utilizatorul către situl Web Macromedia.

ETICHETA `<EMBED>`

Metoda veche pentru a pune un plug-in pe o pagină constă în folosirea etichetei `<EMBED>`. Sunt necesare trei atribute, cu sintaxa:

```
<EMBED src="URL" width=wPixels height=hPixels>
```

În acest caz, `src` este localizarea fișierului plug-in, iar `width` și `height` definesc cât spațiu va ocupa plug-in-ul pe pagina Web. Pentru plug-in-urile care necesită parametri suplimentari, adăugați atribute etichetei `<EMBED>`. Verificați documentația pentru plug-in-ul pe care doriți să-l folosiți ca să vedeți de ce are nevoie.

Pentru a-i ajuta pe utilizatorii care nu folosesc browsere cu capacități plug-in, folosiți eticheta `<NOEMBED>` imediat după `<EMBED>`. Această pereche de etichete creează un spațiu pentru o vedere rapidă a plug-in-ului în acțiune, un text alternativ sau o hiperlegătură către pagina de bază a aplicației. Sintaxa este următoarea:

```
<EMBED src="URL" width=wPixels height=hPixels>
<NOEMBED>
...substitute content...
</NOEMBED>
```

Browserele compatibile cu plug-in-urile ignoră tot ce se află între etichetele `<NOEMBED>`. Dacă un browser nu recunoaște eticheta `<EMBED>`, nu va recunoaște eticheta `<NOEMBED>` și nici nu va afișa nimic din ceea ce se află între granițele sale.

Folosirea altor obiecte și a caracteristicilor speciale

Ca un supliment la toate lucrurile interesante posibil de realizat cu eticheta `<OBJECT>`, mai sunt alte câteva opțiuni pentru adăugarea interactivității și a caracteristicilor speciale paginii dumneavoastră Web. Multe dintre aceste articole sunt de proprietar, deci funcționează doar pe unul sau două browsere.

Fișiere GIF animate

Fișierele GIF animate sunt acceptate indirect în HTML 4.0, prin folosirea etichetelor `<IMG>` și `<OBJECT>` pentru introducerea fișierelor GIF standard. Un fișier GIF animat conține mai multe imagini combinate într-un singur pachet; aceste imagini sunt apoi afișate pe rând pentru a crea o animație simplă.

Există avantaje și dezavantaje ale acestei tehnologii. Vestea bună este că implementarea este mai ușor de realizat decât în cazul lui Java Animator, deși pierdeți componenta de sunet. Vestea proastă este că produce fișiere GIF foarte mari, care au nevoie de mult timp pentru a se descărca. Într-o animație GIF, toate imaginile incluse sunt salvate în același fișier, la un loc cu instrucțiunile care precizează de câte ori vor fi afișate.

Un GIF animat funcționează și cu browserele care nu acceptă partea de animație, fiind afișată doar prima imagine din fișier. Desigur, utilizatorul trebuie să aștepte până se descarcă și celelalte imagini din fișier.

Pentru a crea un GIF animat, aveți nevoie de un set de fișiere imagine și de un instrument pentru a le asambla. Aveți următoarele opțiuni, în funcție de platforma dumneavoastră:

- Windows: Cel mai cunoscut instrument este Alchemy Mindworks GIF Construction Set, care acceptă crearea unor imagini transparente și intertesute. Este disponibil la `www.mindworkshop.com/alchemy/`.
- Macintosh: GIFBuilder este un instrument freeware pe care îl puteți folosi pentru a asambla un set de fișiere GIF, PICT sau TIFF într-un fișier GIF animat. Este disponibil la `www.mid.net/INFO-MAC/`.

■ UNIX: Cu un instrument cu linie de comandă numit whirlGIF, puteți asambla un set de fișiere GIF într-un GIF animat. Include o varietate de opțiuni pentru animare, care sunt explicate la `www.msg.net/utility/whirlgif/`.

În acest moment, probabil că vă întrebați de ce nu puteți pur și simplu folosi vechiul și singurul dumneavoastră editor de imagine. Animația a fost o caracteristică a fișierelor GIF de mai multă vreme, dar nu a fost acceptată în prea multe locuri. Recenta popularitate de care se bucură pe Web a surprins programele grafice nepregătite.

După ce asamblați animația, încercați să o încărcați în browserul dumneavoastră . Dacă nu se întâmplă nimic, asigurați-vă că folosiți ultima versiune de Navigator, Internet Explorer sau Mosaic.

Video cu dynsrc: Internet Explorer

Microsoft a echipat Internet Explorer, prin adăugarea atributului `dynsrc` la eticheta `<IMG>`, pentru a manevra fișierele video AVI. Sintaxa pentru o imagine devine astfel următoarea:

```
<IMG src="imageURL" dynsrs="video.avi">
```

Dacă browserul nu acceptă `dynsrc`, ignoră acest atribut și folosește imaginea identificată în `src`.

Microsoft a adăugat și alte câteva atribute pe lângă `dynsrc` pentru a ajuta la controlul afișării sale, printre care se regăsesc următoarele:

■ **`controls`:** Acest set de controale adăugate în partea de jos a cadrului video oferă funcțiile fast-forward, rewind, stop și play.
■ **`loop`:** Această valoare determină de câte ori trebuie rulat filmul. De exemplu, valoarea `3` cauzează repetarea de trei ori. Valoarea `-1` sau `INFINITE` face ca filmul să fie rulat fără oprire.
■ **`start`:** Sunt două modalități de a porni un film. Prima este `FILEOPEN`, care provoacă filmul să înceapă imediat ce întreaga pagină Web și fișierul AVI au fost încărcate. Cealaltă opțiune este `MOUSEOVER`, care împiedică începerea filmului până când utilizatorul nu trece mouse-ul peste partea de sus a cadrului.

Sunete de fundal: Internet Explorer

Altă opțiune pe care Microsoft a adăugat-o browserelor sale este capacitatea de a rula un fișier de sunet pe post de coloană sonoră a unei pagini. Această caracteristică este posibilă prin eticheta `<BGSOUND>`, care are sintaxa următoare:

```
<BGSOUND src="soundURL" loop=##|INFINITE>
```

În această sintaxă, `src` este un fișier valid de sunet WAV, MIDI sau AU, iar `loop` specifică de câte ori trebuie rulat acesta. Dacă `loop` nu este menționat, va fi rulat o singură dată. Puteți folosi valorile `-1` sau `INFINITE` pentru a rula sunetul fără oprire.

Rezumat

În HTML 4.0, vă puteți aștepta ca eticheta `<OBJECT>` să înlocuiască toate etichetele folosite în prezent pentru inserarea conținutului inline, cum ar fi imagini, sunete, miniaplicații Java, plug-in-uri și controale ActiveX. Eticheta, cum este ea propusă de W3C, include toate atributele necesare acestor articole, astfel încât să mențineți același control asupra înfățișării și comportamentului lor.

Totuși, singurul browser care acceptă în prezent `<OBJECT>` pentru una dintre capacitățile sale este Internet Explorer, care o întrebuințează la implementarea controalelor ActiveX. Dacă încercați să folosiți `<OBJECT>` pentru altceva în IE, va rezulta blocarea programului când acesta va încerca să-și dea seama ce doriți. Navigator ignoră eticheta `<OBJECT>` și trece la conținutul alternativ din cadrul său, la fel cum ar face orice alt browser.

Pe scurt, eticheta cu întrebuințări multiple `<OBJECT>` este o idee bună inclusă în standard, dar nu este încă folosită de browserele favorite. Veți avea nevoie să folosiți eticheta nerecomandată `<APPLET>` și cea nonstandard `<EMBED>` pentru Java și plug-in-uri respectiv, ca o adăugare la alte atribute de proprietar pentru video și sunet inline. Eu vă sugerez să folosiți metodele de proprietar măcar până la următoarea revizuire majoră a browserelor cunoscute.

PARTEA a IV-a

Tehnici de prezentare

Capitolul 17
Stil simplu cu text

de Rick Darnell

În acest capitol

- Formatarea textului cu stiluri fizice 222
- Stiluri fundamentale de formatare a textului: <I>, <B>, <U> 223
- Text barat: <STRIKE> și <S> 224
- Text teletype sau monospațiat: <TT> 225
- Scriere cu superscript și subscript: <SUP> și <SUB> 225
- Lucrul cu literele: <FONT> 226

Formatarea textului cu stiluri fizice

O mare parte din această carte s-a referit la HTML-ul structural, care definește cum sunt utilizate bucățile de text, nu cum ar trebui să apară ele. După cum este recunoscut, există o relație de facto între cele două, cu toate că alegerea finală aparține totuși browserului: un browser marca X poate alege să prezinte <EM> prin majuscule reduse, nu prin cursive. Prima măsură pe care o puteți lua pentru a fi sigur că <EM> este afișat în felul în care doriți este utilizarea unei foi de stil în cascadă, care se tratează în capitolul 18, „Utilizarea foilor de stil în cascadă".

Dacă majoritatea beneficiarilor dumneavoastră nu sunt dotați cu un browser capabil să lucreze cu foi de stil, precum Internet Explorer 3.02 sau Netscape Navigator 4.02, atunci mai dispuneți de o opțiune. Puteți forța stilul prin utilizarea unor etichete HTML speciale.

O dată cu descoperirea limbajului HTML de către designerii de pagini și de către alte persoane cu aplecare spre aspect în Web, au devenit repede vizibile lacunele sale atunci când se ajunge la controlul detaliilor de aspect al paginii. Stilurile logice nu acoperă toate posibilitățile avute în vedere de designerii și autorii de pagini. Designerii au început să strige după un control sporit și au fost recompensați cu etichete și atribute de către Microsoft și Netscape. Folosirea stilurilor fizice permite unui designer de pagină să preia controlul asupra multor detalii de aspect al paginii, incluzând stil de caractere, dimensiune și culoare.

ATENȚIE!

Multe dintre etichetele și atributele menționate în acest capitol au devenit nerecomandate în HTML 4.0, fiind preferate de stil. Am indicat care elemente sunt actuale și care sunt perimate. Documentele construite cu caracteristici nerecomandate vor corespunde numai definiției largi a limbajului HTML, neîncadrându-se în definiția strictă.

ATRIBUTE COMUNE

Cele mai multe elemente din acest capitol au atribute HTML comune, cum ar fi acelea care controlează identificarea, stilul, limbajul și tratarea evenimentului. Întrucât atributele comune sunt atât de răspândite în HTML 4 (de aceea sunt numite „comune"), veți găsi o trecere în revistă a lor în introducerea acestei cărți. De asemenea, puteți găsi mai multe informații despre aceste des folosite caracteristici și elementele asociate lor în Anexa A.

Până una alta, pe parcursul acestui capitol mă voi referi la atributele comune folosind aceste cuvinte cheie: *core (principale), international (internaționale)* și *event (evenimente)*.

Stiluri fundamentale de formatare a textului: <I>, <B>, <U>

Primele trei elemente de formatare a textului reprezintă standarde în aproape toate mediile de procesare a textului. Ele stabilesc aspectul de bază al textului (cum se arată în Figura 17.1) și sunt derivate din multe stiluri logice, cum ar fi cursiv (italice, <I>) pentru text evidențiat (<EM>) și aldin (bold, <B>) pentru text puternic accentuat <STRONG>.

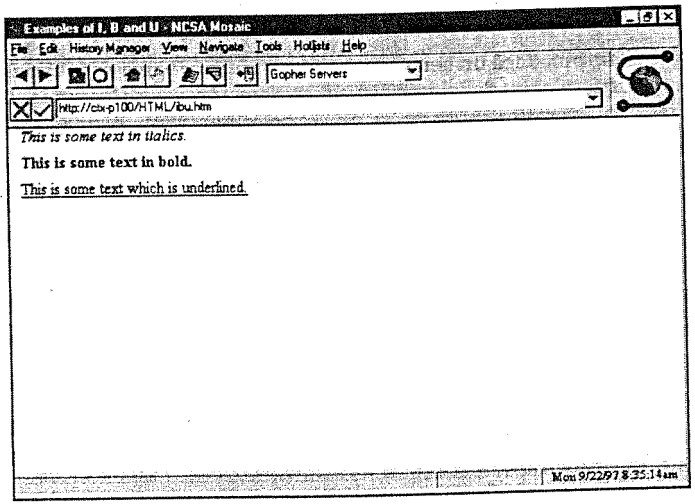


FIGURA 17.1. *Cele trei stiluri fizice de bază sunt cursiv, aldin și subliniat.*

Sintaxa fiecăruia dintre aceste elemente este:

```
<I>...Text...</I>
<B>...Text...</B>
<U>...Text...</U>
```

<I> simbolizează stilul cursiv, <B> stilul aldin, iar <U> pe cel subliniat. Fiecare dintre elemente acceptă, de asemenea, atributele core, international și event.

SUGESTIE

<B> și <I> nu sunt perimate în standardul HTML 4.0, dar <U> este nerecomandat.

Suplimentar utilizării singulare a etichetelor, le puteți imbrica pentru un efect combinat, ca acesta:

```
This is <B>bold text with <I>bold and italics text</I></B>.
```

Din zilele mașinilor tipografice și ale notelor de corector, textul subliniat a fost folosit pentru a indica elementele care ar fi trebuit să apară cu caractere cursive, cum ar fi titlurile de cărți și de ziare. O altă posibilă folosire a textului subliniat a fost introdusă cu eticheta <INS> (vezi Capitolul 7, „Alinierea și formatarea textului"). Eticheta <INS>, folosită pentru a marca textul care a fost inserat în document, este de ajutor în cazul unui document în cadrul revizuirii și editării. Deoarece <INS> nu este acceptată pe scară destul de largă, puteți folosi <U> ca marcaj de siguranță în maniera următoare:

```
<U><INS datetime="1997-09-21T19:38:00Z">This was a good idea and
was inserted.</INS></U>
```

Amintiți-vă că textul subliniat poate crea confuzie pentru cititori în cadrul unei pagini Web, deoarece el este folosit și pentru a marca hiperlegături către alte documente, în special când culoarea textului este diferită de a textului înconjurător.

Variațiuni asupra dimensiunii de bază: <BIG> și <SMALL>

În specificația HTML 4.0 sunt incluse două mici etichete subtile care arată că textul ar trebui să fie mai mare sau mai mic față de dimensiunea prestabilită. Browserul are posibilitatea de a alege dimensiunea de text pe care o crede potrivită, pentru ambele elemente; valoarea lor este relativă la dimensiunea de font de bază a documentului.

Spre exemplu, dacă un beneficiar folosește o dimensiune de font mare (18 pt) pentru a veni în ajutorul celor cu deficiențe de vedere, atunci browserul ar putea alege să redea textul din interiorul etichetei <BIG> cu 24 pt, iar pe cel din <SMALL> cu 14 pt. Sau ar putea alege să ignore ambele etichete.

SUGESTIE

În general, obțineți cele mai bune rezultate cu aceste două etichete când nici un alt stil nu este asociat documentului sau când ele posedă propriile lor definiții de stil pentru a întregi dimensiunea de bază a textului.

Text barat: <STRIKE> și <S>

Textul barat se folosește mai des în procesarea de text tradițională decât în paginile HTML. El indică un text care a fost șters, dar încă este lăsat pe pagină pentru corectare. În acest mod, alți cititori pot revedea modificările și pot ști cum a progresat revizia documentului. Sintaxa poate folosi una dintre formele următoare:

```
<STRIKE>...Text...</STRIKE>
```

```
<S>...Text...</S>
```

Ambele etichete acceptă atributele core, international și event. Scopul tradițional pentru <STRIKE> a fost acoperit de eticheta <DEL>, care este descrisă mai în detaliu în Capitolul 7. Deoarece <DEL> nu este acceptată pe scară destul de largă, puteți folosi totuși <STRIKE> drept marcaj de siguranță în maniera următoare:


```
<STRIKE><DEL datetime="1997-09-21T19:38:00Z">This was a bad idea and
was deleted.</DEL></STRIKE>
```

Atât <STRIKE>, cât și <S> sunt nerecomandate în standardul HTML 4.0.

Text teletype sau monospațiat: <TT>

Mai multe stiluri logice de text beneficiază de avantajul textului monospațiat. Spus simplu, în textul monospațiat, fiecare caracter ocupă aceeași lățime pe ecran. În mod normal, browserele grafice folosesc fonturi de lățime variabilă, ceea ce semnifică faptul că un *m* ocupă mai mult spațiu decât un *i*, care ocupă mai puțin spațiu decât un *j*. Într-un font monospațiat, toate trei au aceeași lățime, la fel ca și celelalte caractere, cum ar fi spațiile, punctele și alte semne de punctuație (vezi Figura 17.2).

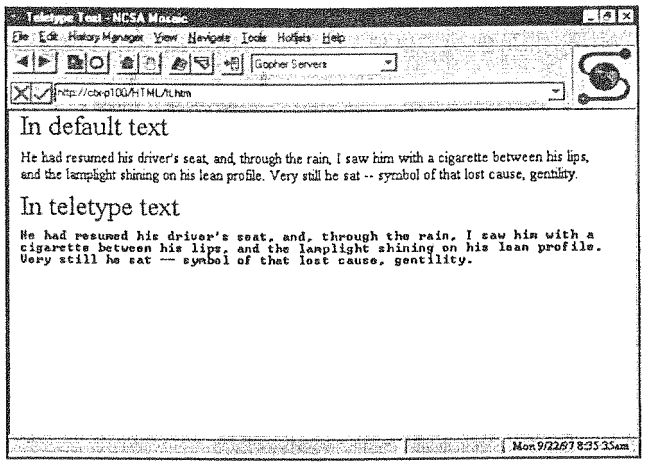


FIGURA 17.2. *Diferența între fonturile normale și cele monospațiate nu este foarte evidentă până ce nu sunt vizualizate unul lângă altul.*

Sintaxa pentru utilizarea etichetei <TT> este aceasta:

```
<TT>...Text...<TT>
```

Text este textul de afișat cu font monospațiat. Această etichetă acceptă atributele core, international și event.

Specificarea literelor TT în etichetă ne duce cu gândul înapoi la zilele teleimprimatoarelor (mașini teletype). În esență, ele erau mașini de scris conectate la rețele care distribuiau știri și buletine. Tipic pentru mașinile de scris ale acelor timpuri, ele foloseau un stil de caractere monospațiat.

Scriere cu superscript și subscript: <SUP> și <SUB>

Următoarele pe lista stilurilor fizice sunt două etichete pentru crearea de superscripturi și subscripturi. Un *superscript* reprezintă text așezat puțin mai sus decât textul de bază, iar un *subscript* este așezat puțin mai jos. Ambele folosesc uzual un tip de caracter mai mic. Iată sintaxa pentru ambele:

```
<SUP>Text</SUP>
```

```
<SUB>Text</SUB>
```

Ambele etichete acceptă atributele core, international și event. În mod normal, superscriptul sau subscriptul este folosit doar pentru scrierea unui caracter sau două. Frag-

mentul următor folosește aceste caracteristici pentru a indica o notă de sfârșit (superscript) și pentru scrierea unei formule chimice (subscript) (vezi Figura 17.3).

```
The chemical formula<SUP>1</SUP> for Glacial Acetic acid
is C<SUB>2</SUB>H<SUB>4</SUB>O<SUB>2</SUB>.
<BR><SUP>1</SUP>Chemical Hazards Response Information System.
```

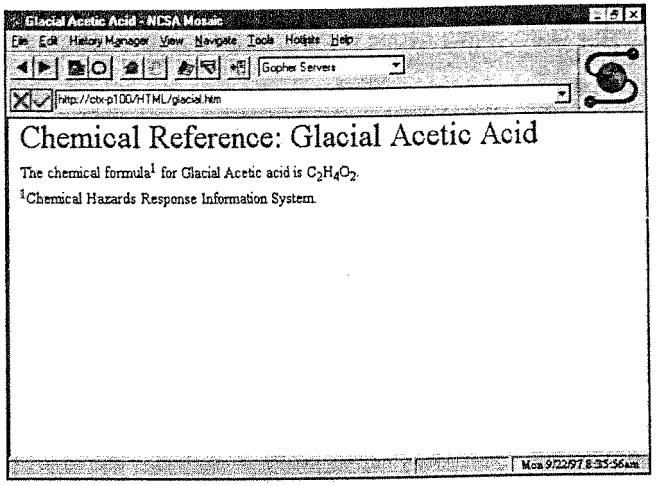


FIGURA 17.3. *Superscripturile și subscripturile sunt folosite în mod obișnuit pentru bucăți mici de text, precum note de sfârșit și alte notații speciale.*

Precum celelalte elemente, puteți imbrica aceste etichete pentru un folos în plus. De exemplu, puteți face ca superscriptul care trimite la o notă de sfârșit să servească și drept legătură spre nota efectivă, astfel:

```
The chemical formula<A HREF="#1"><SUP>1</SUP></A> for Glacial Acetic acid
is C<SUB>2</SUB>H<SUB>4</SUB>O<SUB>2</SUB>.
<BR><A NAME="1"><SUP>1</SUP></A>Chemical Hazards Response Information System.
```

Un clic pe exponentul pentru nota de sfârșit duce beneficiarul la respectiva locație a documentului. Pentru mai multe informații despre lucrul cu ancore și hiperlegături, vezi Capitolul 13, „Hiperlegături și ancore".

Lucrul cu literele: <FONT>

Elementul <FONT> este utilizat pentru a schimba caracteristicile fontului de bază – tip de caracter, dimensiune și culoare – în blocurile de text. Folosirea sa nu este recomandată în HTML 4.0.

Atributul face este utilizat pentru a specifica unul sau mai multe tipuri de caracter pentru selecția de text. Dacă ați dori să specificați tipul de caracter Garamond drept primă variantă și Arial ca a doua, eticheta ar arăta ca mai jos:

```
<FONT face="Garamond,Arial">...text...</FONT>
```

Dacă nici unul dintre tipurile de caracter specificate nu este găsit pe browserul utilizatorului, este folosit tipul de caracter prestabilit pentru respectiva aplicație particulară. Dacă doriți să folosiți acest atribut, asigurați-vă că pagina dumneavoastră va fi afișată corect chiar dacă browserul utilizatorului nu îl acceptă.

Stabilirea dimensiunii: size și <BASEFONT>

Un atribut important pentru <FONT> este size. El stabilește dimensiunea conținutului etichetei la o valoare specifică sau relativă. Sintaxa este aceasta:

```
<FONT size=number>...Text...</FONT>
```

Aici, *number* este dimensiunea dorită exprimată ca un întreg de la 1 la 7 sau ca o valoare relativă de la -6 la +6. Dacă se folosește o valoare relativă, ea se adaugă valorilor curente pentru <BASEFONT>. (Această metodă este tratată mai târziu în secțiunea de față.) Spre deosebire de titluri, unde <H1> reprezintă cel mai mare titlu, iar <H6> este cel mai mic, atributul size folosește 1 pentru dimensiunea celui mai mic tip și 7 pentru cea mai mare (vezi Figura 17.4).

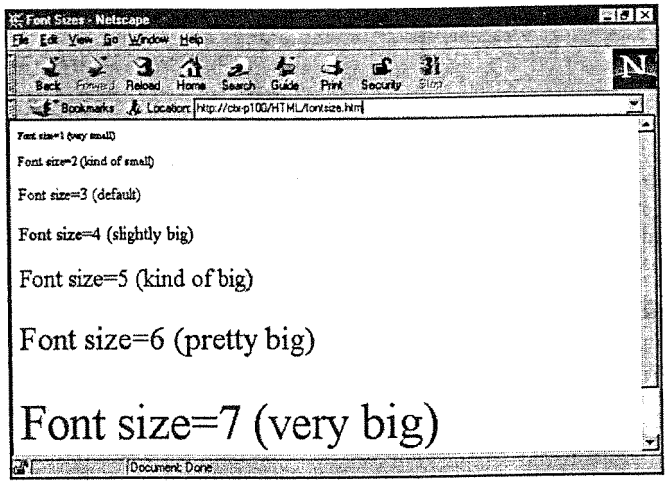


FIGURA 17.4. *Modul în care sunt reprezentate diverse dimensiuni de text folosind elementul FONT.*

Pentru persoanele obișnuite să lucreze cu dimensiunile tipului de caracter reprezentate în puncte tipografice, intervalul dimensiunilor pentru text în HTML nu pare practic. În definitiv, caracterele de 1 punct sunt atât de mici, încât nici măcar un avocat nu le-ar înțelege.

Dimensiunile de caracter HTML chiar corespund dimensiunilor fizice, dar mărimea fizică reală variază după browsere și platforme. Deci nu considerați că dimensiunea 1 produce un text de 6 puncte. Mai bine gândiți dimensiunea 1 ca „într-adevăr mică", dimensiunea 7 ca „într-adevăr mare", iar 2 până la 6 ca un interval între „destul de mic" și „destul de mare".

Ca un însoțitor al atributului size pentru <FONT>, elementul BASEFONT vă oferă un mod de a stabili dimensiunea de bază a textului de rând. Iată sintaxa sa:

```
<BASEFONT [size=number]>
```

În acest element, *number* este dimensiunea fontului de bază, exprimată ca un întreg de la 1 la 7. Dacă este omisă, se folosește valoarea prestabilită 3. Ea fixează dimensiunea de bază pentru tot textul care o urmează și nu are o etichetă de închidere.

Pe lângă specificarea unei dimensiuni de text absolute, puteți folosi și un număr relativ pentru <BASEFONT>. Această metodă utilizează dimensiunea de font prestabilită (3), la care adaugă valoarea atributului size (vezi Figura 17.5).

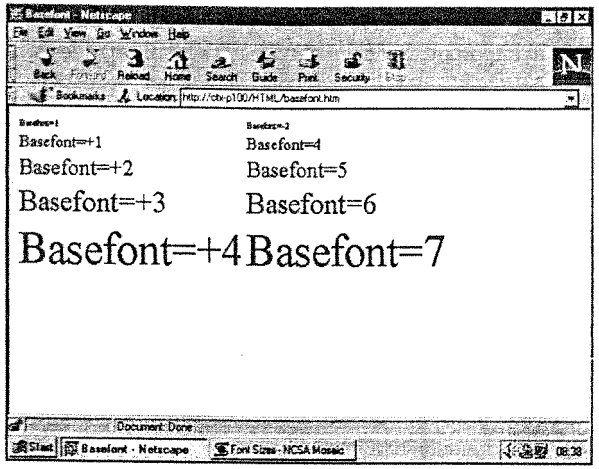


FIGURA 17.5. *Indiferent cât este dimensiunea fontului anterior, o valoare relativă pentru fontul de bază este adăugată totdeauna dimensiunii prestabilite.*

Folosind dimensiuni relative, șirul valorilor pentru atributul `size` de bază este de la `-2` la `+4`. Atât pentru `<BASEFONT>`, cât și pentru `<FONT>`, atunci când valorile relative au ca rezultat dimensiuni mai mici decât `1` sau mai mari decât `7`, se folosește valoarea minimă, respectiv maximă a dimensiunii fizice.

O umbră diferită de text: `color`

Puteți controla culoarea unui bloc de text folosind atributul `color`, care are următoarea sintaxă:

```
<FONT color=name|RGBValue>...Text...</FONT>
```

Aici, valoarea pentru `color` este un *name* (numele uneia dintre cele 16 culori standard) sau un triplet hexazecimal RGB care specifică amestecul de culori de utilizat. După cum vă amintiți, culorile prestabilite pentru textul din întregul document sunt precizate cu atributele etichetei `<BODY>`, care a fost tratată într-un capitol anterior.

OBSERVAȚIE

Puteți defini o culoare în două feluri. Primul este tripletul hexazecimal RGB, care este foarte exact, dar nu prea ușor de reamintit.

Sau puteți folosi numele culorii. Este mult mai ușor, dar nu puteți folosi chiar oricare nume. De exemplu, chiar mi-a plăcut culoarea automobilului prietenului meu John. Era taupe (gri-maroniu). Din nefericire, nu pot tasta `<FONT color="taupe">` și să mă aștept ca browserul să înțeleagă.

Dacă doriți să folosiți nume de culori, veți avea nevoie să vă adaptați uneia dintre cele 16 valori standard: black (negru), green (verde), silver (argintiu), lime (vernil), gray (gri), olive (măsliniu), white (alb), yellow (galben), maroon (castaniu), navy (bleumarin), red (roșu), blue (albastru), purple (purpuriu), teal (albastru-verzui), fuchsia (roșu-purpuriu) și aqua (turcoaz). Unele browsere folosesc alte nume. Pentru a fi sigur, folosiți unul dintre aceste nume sau scoateți un calculator științific și calculați 32 procente de albastru ca un număr hexazecimal (este #000020).

Aveți grijă la schimbarea culorilor textului, în special când folosiți un fundal neprestabilit, precum o imagine sau o altă culoare. Unele culori nu se împacă împreună, cum ar fi text verde pe roșu sau text roșu pe purpuriu. Efectul rezultat îi va deranja pe cititorii dumneavoastră în loc să vă ilustreze punctul de vedere.

Rezumat

Etichetele din acest capitol reprezintă într-adevăr un mod simplu de a adăuga stil paginii dumneavoastră Web. Prin inserarea rapidă a uneia sau două etichete, puteți cu ușurință să modificați aspectul unui bloc de text. Totuși, majoritatea lor nu mai sunt recomandate în HTML 4.0, datorită foilor de stil.

Printre elementele care rămân actuale în HTML 4.0 se numără următoarele:

`TT`	Teletype (teleimprimator)
`I`	Italic (cursiv)
`B`	Bold (aldin)
`BIG`	Mai mare decât cel prestabilit
`SMALL`	Mai mic decât cel prestabilit
`SUP`	Superscript (exponent)
`SUB`	Subscript (indice)

Printre elementele care nu sunt recomandate în HTML 4.0 se numără următoarele:

`STRIKE, S`	Barat
`U`	Subliniere
`FONT`	Tipul de caracter, dimensiunea și culoarea
`BASEFONT`	Dimensiunea de text prestabilită

Deși oricare dintre etichetele din aceste două liste pot fi folosite pentru a ajusta fin aspectul textului dumneavoastră, obțineți cele mai bune rezultate cu foile de stil, care sunt tratate în capitolul următor.

Capitolul 18
Utilizarea foilor de stil în cascadă

de Rick Darnell

În acest capitol

- **Controlul stilului prin foi de stil** 230
- **Organizarea stilului** 247
- **Pretutindeni foi de stil: o culegere de exemple** 249

În capitolul anterior, ați avut șansa de a schimba aspectul unui document utilizând etichete HTML pe bucăți de text. Această metodă se folosește, dar nu este în mod necesar cea mai ușoară. Mai mult, ori de câte ori doriți să reproduceți un text roșu, sunteți nevoit să încadrați textul cu eticheta `<FONT text="red">`.

Foile de stil în cascadă oferă instrumentele de a rafina paginile dumneavoastră Web până la al n-ulea grad, fără să se piardă compatibilitatea cu browserele mai vechi. CSS (Cascading Style Sheet – foi de stil în cascadă) este un standard acceptat de Consorțiul World Wide Web.

În esență, foile de stil reglementează modul în care sunt formatate etichetele HTML. Rândurile autorilor Web sunt tot mai populate de oameni din domeniile designului grafic sau al tehnoredactării computerizate, iar a controla diferențele fine de stil devine parte a apropierii naturale de design. Totuși, primele versiuni de HTML nu ofereau un instrument comun pentru controlul stilului, iar dezvoltările ulterioare conțineau unul limitat.

Deci, cu toată această introducere, încă nu am stabilit ce este CSS. Cuvântul *în cascadă* înseamnă că pot fi utilizate stiluri multiple într-o pagină individuală HTML, iar browserul va urma o ordine – numită *cascadă* – spre a interpreta informațiile. Designerul poate utiliza toate cele trei tipuri de foi de stil simultan, iar browserul va livra marfa într-un mod ordonat, predictibil. Foile de stil în general, și CSS în particular, sunt o importantă metodă de a controla modul în care textul este afișat și mutat în cadrul paginii, iar acest capitol vă arată cum se face.

Controlul stilului prin foi de stil

De-a lungul evoluției limbajului HTML, controlul stilului a fost greoi în cel mai fericit caz și inexistent la polul opus. Totuși, prin tehnica foilor de stil în cascadă, proiectantul Web și-a îmbunătățit substanțial modalitățile de lucru cu elemente de stil importante, precum fonturile, și cu anumite trăsături, precum aspectul fontului, mărimea fontului, grosimea sa, stilul său și interlinierea. Controalele de stil se extind dincolo de funcțiile tipografice de bază, asupra marginilor, alinierilor, culorilor, graficii și a unei miriade de alte opțiuni.

Consorțiul World Wide Web (W3C) a emis un standard definitiv pentru foi de stil, care este în prezent reflectat în produsul Microsoft Internet Explorer 3.0 și versiunile ulterioare, precum și în produsul Netscape Navigator 4.0 și versiunile ulterioare. Cum cele două browsere principale au acum aceste implementări, vă puteți aștepta ca foile de stil să devină o parte integrantă a multor pagini Web.

OBSERVAȚIE

Când o pagină care utilizează exclusiv foi de stil este vizualizată cu versiunile mai vechi de Netscape (3.0 sau anterioare) sau Internet Explorer (2.0 sau anterioare), pagina Web revine la fundalul gri de temniță. Browserele mai vechi retrogradează fonturile și alte stiluri la valorile lor prestabilite, elementare (vezi Figurile 18.1 și 18.2).

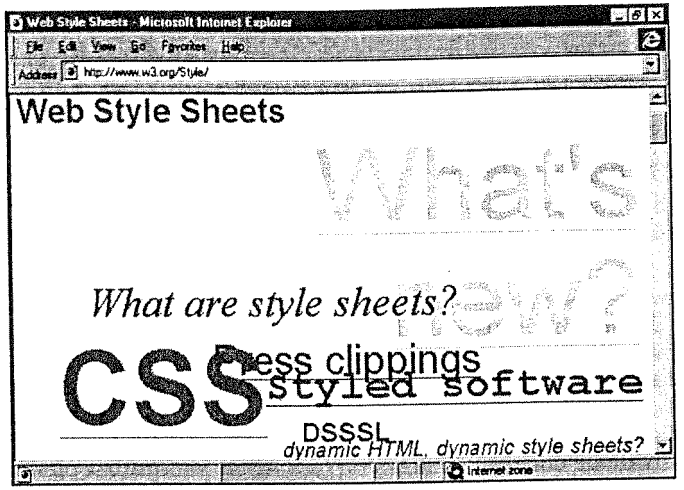


FIGURA 18.1. *O pagină care utilizează foi de stil în cascadă așa cum apare în Internet Explorer 4.0, prezentată în toată strălucirea sa.*

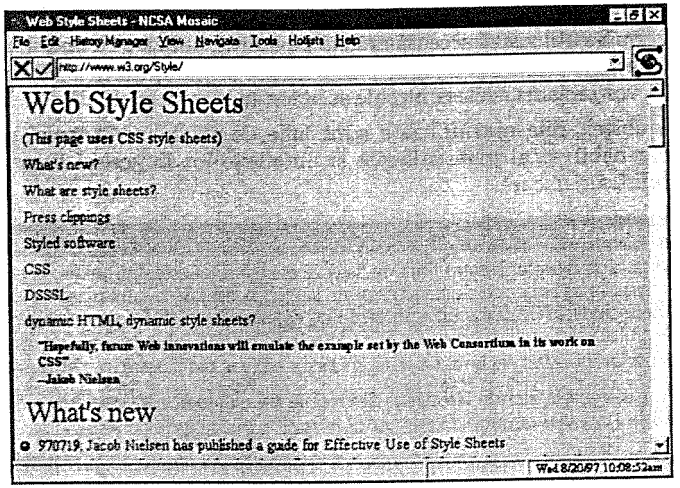


FIGURA 18.2. *Aceeași pagină revine la aspectul prestabilit, elementar într-un browser fără caracteristica de control prin foaie de stil, cum ar fi Mosaic.*

Cu toate promisiunile lor, există încă unele probleme în utilizarea foilor de stil, precum diferența dintre fonturile pe care fiecare utilizator le alege pe mașina sa. Alegerea mea de fonturi diferă de a dumneavoastră, selecția dumneavoastră diferă de a colegului, iar alegerea colegului dumneavoastră diferă de cea a vecinului său. O foaie de stil poate cere un anume aspect de literă, dar dacă fontul și grosimea sa nu există pe mașina oaspete, browserul nu poate interpreta stilul pe care l-ați prestabilit.

Și, precum s-a menționat în observația anterioară, o pagină care folosește exclusiv foi de stil nu va da rezultatele dorite într-un browser necompatibil. Dacă nu se ține cont, într-o anumită măsură, că există și multe alte tipuri de browsere de Web, o pagină poate arăta ca și cum ar fi fost creată într-adins urâtă și de necitit.

Din fericire, există ocolișuri pentru probleme. Designerii înțelepți adună informațiile în șirurile de argumente ale stilului și îl salvează cu atribute în etichete HTML, ceea ce permite browserului să descopere informațiile pe care le poate interpreta. Acesta este un lucru bun, dar designerii nu dispun de un control total, deoarece nu pot fi siguri că o pagină va arăta la fel indiferent unde este vizualizată. Rămâne adevărată vechea regulă: fiți prudent și atent și, ori de câte ori este posibil, testați rezultatele muncii dumneavoastră prin vizualizare cu o varietate de browsere pe o multitudine de platforme.

Foi de stil de bază

Există trei moduri fundamentale de a utiliza foile de stil: metoda inline, metoda paginii individuale (a înglobării) și legarea la o foaie de stil master (externă).

- **Foaie de stil inline:** Aceasta beneficiază de avantajul etichetelor HTML existente într-un document HTML standard și adaugă un anume stil informațiilor controlate de acele etichete. Un exemplu de utilizare a foilor de stil inline este cel de control al indentării unui singur paragraf prin utilizarea atributului `style="someStyle"` în interiorul etichetei `<P>`. O altă metodă de a face aceasta este prin combinarea etichetei `<SPAN>` cu atributul `style="someStyle"`.
- **Foaie de stil înglobat:** Prin această metodă, designerul poate controla pagini individuale utilizând eticheta `</STYLE>`. Aceste informații se plasează între etichetele `<HTML>` și `<BODY>`; atributele de stil sunt inserate în interiorul containerului `<STYLE>`.
- **Foaie de stil extern (legat):** Aceasta este o foaie de stil master memorată într-un fișier extern. Se utilizează aceeași sintaxă pe care ați utiliza-o cu un stil înglobat. Acest fișier folosește extensia `.css`. Apoi, pur și simplu asigurați-vă că toate documentele HTML care vor reclama acele controale sunt legate la respectivul document.

De obicei, cele mai utilizate sunt foile de stil inline și înglobat deoarece este foarte convenabil să ai toate stilurile și atributele lor în același fișier, pentru o revedere și modificare ușoară.

OBSERVAȚIE

Sintaxa atributelor pentru foile de stil înglobat și legat este, într-un fel, diferită de sintaxa standard HTML. În cazul controlului prin foi de stil (CSS), atributele sunt plasate în interiorul acoladelor. Acolo unde HTML plasează un semn de egalitate (=), CSS folosește două puncte (:). În HTML atributele individuale sunt separate de spațiu, iar CSS folosește punct și virgulă. Un șir al foii de stil poate arăta astfel: `{font-style: arial, helvetica; margin-left}`.

Totuși, ca și în cazul HTML, sintaxa foii de stil este logică. Dacă lucrați cu exemplele din acest capitol, vă veți familiariza cu modul în care lucrează foile de stil.

Când vă referiți la atributele stilului ca proprietăți ale unui obiect, cratima dispare și prima literă a celui de al doilea cuvânt devine majusculă. Astfel, `font-style` devine `fontStyle`, iar `text-decoration` devine `textDecoration`.

Următoarele paragrafe ar trebui să servească drept introducere pentru începători și drept recapitulare pentru bătrânii cunoscători ai fiecărui tip de foaie de stil. Pentru a crea exemplele următoare au fost utilizate unele elemente de sintaxă a foii de stil, dar sintaxa detaliată este tratată mai târziu în acest capitol și în anexa C.

Utilizarea stilurilor inline

Un stil inline este inclus în eticheta pe care o afectează. El nu contează pentru celelalte etichete HTML de pe pagină, fie că ele sunt sau nu de același tip. De exemplu, dacă plasați un stil cursiv într-o etichetă `<HT>`, stilul nu se va aplica celorlalte etichete `<HT>` de pe pagină. Un stil inline este creat prin adăugarea atributului `style` fiecărui element afectat.

Pentru a crea un stil inline, începeți cu o pagină goală în editorul dumneavoastră HTML.

OBSERVAȚIE

Nu am de gând să fac nici o prezumție despre ce editor HTML utilizați. Există o mulțime dintre care să alegeți și îl veți utiliza pe cel cu care sunteți familiarizat. Există totuși câteva cerințe.

În primul rând, ar trebui să admită etichete nestandard, precum `<STYLE>`. Unele browsere validează automat limbajul HTML, unele nu, iar acelea care o fac diferă prin modul în care tratează articolele nestandard. Dacă editorul dumneavoastră nu acceptă direct HTML 3.2, HTML 4.0 sau CSS1 (Cascading Style Sheets Level 1, adică Foi de stil în cascadă, nivelul 1), asigurați-vă că vă permite cel puțin să inserați articole care nu sunt standard în versiunile HTML timpurii.

A doua cerință derivă din prima. Dacă aveți de gând să adăugați elemente nestandard, veți avea nevoie de o modalitate prin care să modificați codul sursă al documentului. Mai mult decât a adăuga noi etichete, veți avea nevoie și de o modalitate de a adăuga atribute etichetelor existente.

Dacă editorul dumneavoastră nu acceptă acest soi de activitate și nu puteți găsi un editor care vă place, atunci puteți totdeauna să apelați la bătrânul dumneavoastră editor de texte. Este posibil să nu fie încântător, dar funcționează.

În continuare, includeți liniile următoare de cod HTML. Acordați atenție atributelor de stil suplimentare din etichetele `<BODY>`, `<P>` și `<SPAN>`.

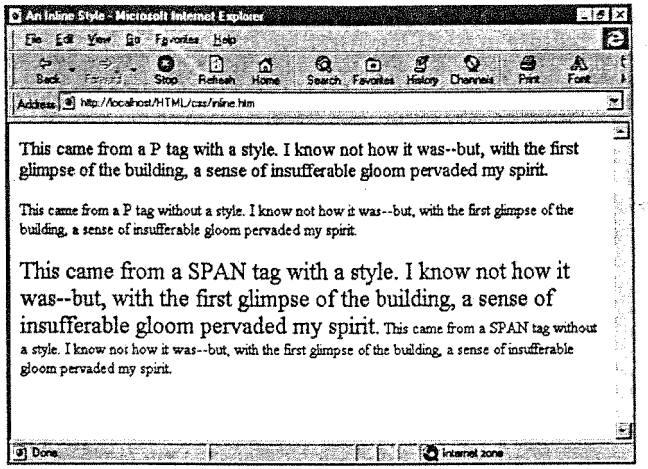
```
<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 4.0//EN">
<html>
<head>
<title>An Inline Style</title>
</head>
<body style="background: white">

<p style="font-size: 14pt" id="p14">
This came from a P tag with a style.
I know not how it was--but, with the first glimpse of the building, a sense of
insufferable gloom pervaded my spirit.</p>

<p>
This came from a P tag without a style.
I know not how it was--but, with the first glimpse of the building, a sense of
insufferable gloom pervaded my spirit.</p>

<p>
<span style="font-size: 18pt" id="span18">
This came from a SPAN tag with a style.
I know not how it was--but, with the first glimpse of the building, a sense of
insufferable gloom pervaded my spirit.</span>

<span>
This came from a SPAN tag without a style.
I know not how it was--but, with the first glimpse of the building, a sense of
insufferable gloom pervaded my spirit.</span></p>

</body>
</html>
```

Salvați fișierul sub numele `inline.html`.

Figura 18.3 arată pagina terminată, așa cum apare în Internet Explorer 4.0.

Precum puteți vedea din exemplu, stilul inline pentru o etichetă nu are nici un efect asupra oricărei alte etichete din împrejurimi. Efectul său este local.

FIGURA 18.3. *Rezultatele comenzilor foii de stil inline.*



SFAT

Eticheta `<DIV>` (diviziune) poate fi folosită ca și eticheta `<SPAN>` pentru controlul inline. Eticheta `<DIV>` este utilă în special pentru blocuri lungi de text, iar `<SPAN>` este deosebit de eficientă pentru a atașa stil la întinderi mai reduse de informații, precum propoziții, câteva cuvinte ori chiar litere aparte din interiorul unui cuvânt.

Într-un fel, un stil inline anulează scopul final al foilor de stil în cascadă. Principalul scop al acestei tehnologii este de a controla stilul pentru o întreagă pagină sau pentru seturi de pagini. Din acest motiv, metoda inline este utilizată obișnuit numai acolo unde sunt necesare unele mici retușuri ale stilului. În plus, atașarea unui stil fiecărei etichete reclamă multă muncă de tastare, astfel că poate genera o grămadă de erori.

Un stil înglobat pentru o pagină

Următorul stil cu care veți lucra este un stil înglobat. O *foaie de stil înglobat* este un set de definiții de stil plasate în interiorul etichetelor `<STYLE>` și găsite între antetul și corpul documentului. Ea stabilește atributele de stil pentru întreaga pagină unde este situată și creată urmând pașii de mai jos:

Începeți cu o pagină goală în editorul dumneavoastră HTML. Tastați textul care urmează. Asigurați-vă că plasați etichetele `<STYLE>` în locul potrivit: imediat după `<HTML>` și înainte de `<BODY>`. Stilul este similar celui tocmai văzut de dumneavoastră la stilurile inline. Sintaxa exactă este descrisă mai în detaliu de-a lungul capitolului.

```
<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 4.0//EN">
<html>
<head>
<title>An Embedded Style Sheet</title>
</head>
```



```
<style>
BODY {background:#0000FF; color: ##FFFF00; margin-left:0.5in; margin-right:0.5in}
H2 {font-size:18pt; color:#FF0000; background:#FFFFFF}
P {font-size:12pt; text-indent:0.5in}
</style>

<body>

<h2 id="h2_1">This heading in red on a white background.</h2>

All text is yellow on a blue background.<BR>

This text is not indented, except for the margins. During the whole
of a dull, dark, and soundless day in the autumn of the year, when the clouds
hung oppressively low in the heavens, I had been passing alone,
on horseback, through a singularly dreary tract of country; and at length
found myself, as the shades of the evening drew on, within view of the
melancholy House of Usher.

<p id="p_1">
This text is indented. During the whole of a dull, dark, and soundless day
in the autumn of the year, when the clouds hung oppressively low in the
heavens, I had been passing alone, on horseback, through a singularly
dreary tract of country; and at length found myself, as the shades of the
evening drew on, within view of the melancholy House of Usher.</p>

</body>
</html>
```

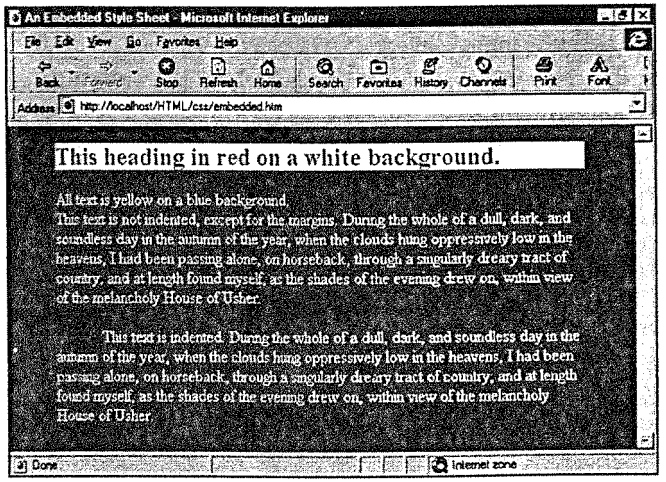
Salvați fișierul sub numele `embedded.html`.

OBSERVAȚIE

Textele pentru exemplificările din acest capitol au fost luate din „The Fall of the House of Usher" de Edgar Allan Poe, așa cum a fost retipărită de Proiectul Gutenberg. Proiectul Gutenberg este dedicat scanării, transcrierii și altor metode de convertire într-un text electronic simplu a cărților din aria publică și a altor lucrări scrise. Puteți accesa textul complet al operei lui Poe, precum și o mulțime de alți clasici în pagina proiectului, `www.promo.net/pg/`.

Figura 18.4 arată pagina finalizată, așa cum apare în Internet Explorer 4.0.

FIGURA 18.4 *Rezultatele comenzilor foii de stil individuale sau înglobate.*



OBSERVAȚIE

Se prea poate să fi observat de acum că utilizez etichetele pentru paragraf drept containere, cu începutul <P> și sfârșitul </P>. Aceasta ține de standardul HTML 4, chiar dacă nu este obligatoriu. Totuși, dacă aveți de gând să lucrați cu foi de stil în cascadă, este nevoie să vă intre în reflex să utilizați etichete de închidere pentru elemente ca paragrafe, celule de tabel și alte articole similare. O foaie de stil nu va recunoaște un paragraf nou până ce nu îl individualizați prin <P> și </P>. Aceasta se aplică chiar dacă este vorba de un paragraf nou, precum prima linie care urmează unui titlu.

Când veți începe să utilizați foi de stil, probabil că veți utiliza în special forma înglobată. Controlul său pagină cu pagină vă permite să modificați aspectul și comportamentul fiecărei pagini a sitului. Totuși, dacă se cere o uniformitate deosebită pentru un set de pagini sau pentru întregul sit, devine imperioasă o legare la o foaie de stil master.

O foaie de stil în cascadă legată

Este o lucrare într-adevăr simplă să creați o foaie de stil care poate fi legată la și utilizată de către oricâte alte pagini. Se numește și *legată* deoarece toate definițiile de stil sunt așezate într-un singur fișier, iar pagina HTML curentă creează o legătură la ea când este încărcată. A crea un stil legat este asemănător cu a crea un stil înglobat. Începeți cu o pagină goală în editorul dumneavoastră HTML.

OBSERVAȚIE

Multe editoare înglobează automat etichetele de structură HTML elementare când creați un document nou, cum ar fi etichetele <HTML>, <HEAD> și <BODY>. Deoarece o foaie de stil legată nu reclamă cerințe anume, veți constata că este mai ușor să folosiți un editor de text.

Plasați definițiile de stil între etichete în același fel în care ați făcut-o pentru foaia de stil înglobat:

```
BODY {background:#0000FF; color: ##FFFF00;
➥margin-left:0.5in; margin-right:0.5in}
H2 {font-size:18pt; color:#FF0000; background:#FFFFFF}
P {font-size:12pt; text-indent:0.5in}
```

Acum, în loc de a vă salva munca într-un fișier HTML, salvați-o cu extensia `.css`. Pentru acest exemplu, numele integral este `master_style.css`.

Acum, integrați fișierul de stil master într-o pagină HTML. Repetați acest proces pentru toate paginile care utilizează aceleași definiții de stil. Pentru a realiza aceasta, întâi selectați pagina HTML căreia doriți să îi aplicați stilul master. Pentru moment, folosiți un fragment al paginii din exemplul foii de stil înglobate:

```
<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 4.0//EN">
<html>
<head>
<title>A Linked Style Sheet</title>
</head>

<body>

<h2>This is a heading.<h2>

It was a mistery all insoluble; nor could I grapple with the shadowy
fancies that crowded upon me as I pondered.
```



```
<p>
I was forced to fall back upon the unsatisfactory conclusion, that while, beyond
doubt, there are combinations of very simple natural objects which have the power
of thus affecting us, still the analysis of this power lies among considerations
beyond our depth.</p>

</body>
</html>
```

Deschideți fișierul cu editorul dumneavoastră HTML, iar apoi plasați linia următoare în interiorul etichetei <HEAD>, după <TITLE>:

```
<link rel=stylesheet href="master_style.css" type="text/css">
```

Eticheta <LINK> identifică alt fișier care este în raport cu pagina curentă. De asemenea, este utilizată pentru a crea butoane de navigare în browsere ca Mosaic. În acest caz, ea identifică o foaie de stil în cascadă din același director cu pagina Web. Dacă aceasta era o foaie de stil accesibilă JavaScript (JASS), tipul său ar trebui să fie `text/jass`. Crearea stilurilor cu JASS este tratată în Capitolul 21, „Folosirea foilor de stil JavaScript".

Antetul complet arată acum astfel:

```
<head>
<title>A Linked Style Sheet</title>
<link rel=stylesheet href="master_style.css" type="text/css">
</head>
```

Salvați fișierul sub numele `linked.html`.

Toate etichetele afectate din pagina Web vor fi interpretate cu referire la stilurile pe care le-ați stabilit de aici înainte în fișierul `.css`. Rezultatele paginii de exemplificare a folosirii unei foi externe de stil legat sunt arătate în Figura 18.5. Pagina nu conține nici o altă informație supra foii de stil, în afara locației fișierului `.css`.

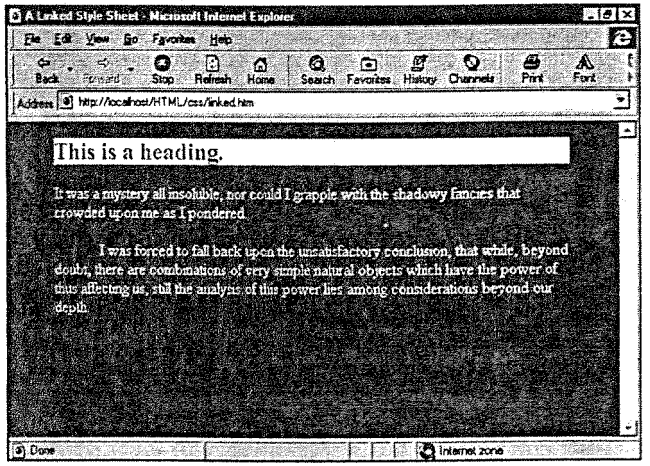


FIGURA 18.5. *Această pagină include o legătură către o foaie de stil externă.*

OBSERVAȚIE

Unele servere Web reclamă înregistrarea tipului MIME pentru foile de stil, înainte ca serverul să poată livra browserului foaia de stil legată. Administratorul Web are nevoie să cunoască faptul că sufixul este `.css` și tipul MIME este `text/css`.

Având informațiile despre stil în fișierul lor propriu, le puteți utiliza pe orice număr de pagini prin simpla adăugare a etichetei <LINK> în antetul documentului.

Combinarea tehnicilor foii de stil

Uneori, soluția cu o singură foaie de stil nu este suficientă, astfel că designerii cu cerințe complexe recurg la mixarea tehnicilor foii de stil. Mixarea folosește din plin caracteristica de aranjare în cascadă a foilor de stil. Tehnicile multiple sunt rezolvate de către browser în felul următor:

- În primul rând, foile de stil legat sunt aplicate global.
- Dacă, pe aceeași pagină, o foaie de stil înglobat se întâlnește cu o foaie de stil legat, atunci stilul înglobat anulează stilul legat, dacă există un conflict.
- Stilurile inline anulează atât stilul înglobat, cât și stilul legat.

Această implementare sistematică și predictibilă vă permite să definiți o singură foaie de stil pentru un sit întreg, să modificați pagini individuale spre a accepta sarcini specifice cu un stil înglobat și să folosiți stiluri inline pentru a pune la punct detaliile unor linii specifice.

OBSERVAȚIE

Dacă situl dumneavoastră include o multitudine de pagini, stiluri legate și stiluri înglobate, este ușor ca la un moment dat să nu mai știți ce se utilizează în fiecare loc și unde este anulat vreun stil. Pentru a vă face viața mai ușoară, mențineți stilurile legate la un nivel cât mai general posibil și apoi aplicați alte tehnici ale foii de stil doar atunci când acestea sunt cerute spre a crea efecte speciale, precum formatarea tabelelor sau stabilirea marginilor pentru un set de imagini.

SUGESTIE

Sintaxa foii de stil este pretențioasă: totul trebuie să fie la locul potrivit pentru a funcționa bine. Dacă pagina dumneavoastră nu afișează stilul pe care l-ați specificat, verificați sintaxa cu atenție. O acoladă, semnul : (două puncte), semnul ; (punct și virgulă) sau alte semne de punctuație plasate greșit în definiția lui ar putea face stilul dumneavoastră nefolositor. Dacă nu găsiți vreo problemă în sintaxă, atunci uitați-vă după probleme cu eventuale tabele, precum etichete de închidere pentru toate rândurile, celulele și alte elemente.

Atribute de stil specifice textului

Aspectul Webului nu se poate modifica în bine dacă designerii nu înțeleg câteva principii de bază legate de text înainte de a porni să utilizeze foile de stil. Unii își dedică întreaga carieră studiului topografiei. Este, incontestabil, o artă, cerând o bună cunoaștere nu doar a nenumăratelor aspecte ale fonturilor, dar și a atributelor asociate, precum grosimea, variațiile de dimensiune, stilurile, familiile, și a modului în care pot fi utilizate artistic toate aceste aspecte în combinații atractive.

Vestea bună este că nu trebuie să petreceți o viață întreagă învățând toate aceste lucruri. Tot ceea ce vă trebuie sunt câteva fire călăuzitoare care să mențină pagina dumneavoastră atractivă și bine scrisă.

Următoarele atribute de text și asociate textului sunt disponibile pentru a fi utilizate cu toate metodele foilor de stil:

- **font-family**: Acest atribut controlează aspectul fontului prin stabilirea argumentelor pentru oricare dintre următoarele:

Numele tipului de caractere pe care doriți să îl utilizați. Iată un exemplu de specificație de stil pentru Times:


```
{font-family: times}
```

Întrucât un calculator particular poate să nu aibă instalat fontul preferat de dumneavoastră, ați putea dori să adăugați o variantă similară:

```
{font-family: times, garamond}
```

Dacă un nume de font din definiția dumneavoastră de stil reprezintă mai mult de un cuvânt, precum Goudy Old Style, va fi nevoie să îl încadrați între ghilimele, precum `font-family: times "goudy old style", garamond`. Altfel, ghilimelele nu sunt necesare.

SUGESTIE

Majoritatea calculatoarelor, inclusiv toate cele cu Microsoft Windows, vin echipate cu fonturi standard, cum ar fi Times și Helvetica. Ele ar putea avea diverse nume, ca Dutch sau Arial, dar reprezintă, în esență același lucru. Puteți găsi mult mai multe tipuri de caractere pe CD-uri ieftine sau puteți obține fonturi de calitate profesională de la companii ca BitStream (`www.bitstream.com`) sau Image Club (`www.imageclub.com`).

Ca o ultimă soluție, puteți adăuga un nume generic de familie, care anunță browserul să aleagă un aspect al literei care, cel puțin, are aceleași atribute de bază ca și celelalte opțiuni ale dumneavoastră:

```
{font-family: serif}
```

Următoarele familii de caractere sunt recunoscute în foile de stil:

serif Fonturile serif (cu serife) reprezintă o alegere bună pentru porțiunile mari de text. Printre cele mai obișnuite se numără Times New Roman și Garamond.

sans serif Această familie de fonturi include versiuni obișnuite, precum Arial, Helvetica și Avant Garde. Fonturile sans serif (fără serife) sunt utilizate adesea pentru titluri spre a întregi imaginea unui text cu serife.

cursive Acestea sunt fonturi de manuscris, care arată de parcă au fost scrise cu mâna. Includ pe unele ca Zapf Chancery și Park Avenue.

fantasy Fonturile fantasy sunt decorative și utile pentru titluri stilate. Însă de obicei ele nu sunt practice pentru textul de rând sau titlurile lungi. Exemplele includ Blippo, Hobo și Broadway.

monospace Această familie de caractere reprezintă o întoarcere la zilele mașinilor de scris și ale teleimprimatoarelor. *Monospace* (monospațiat) înseamnă că fiecare literă ocupă același spațiu: un *i* ocupă la fel de mult ca un *m*. Cele mai multe fonturi sunt proporționale, ceea ce înseamnă că fiecare literă ocupă un spațiu care este proporțional cu dimensiunea și stilul literei. La fonturile proporționale, un *i* ocupă cam o pătrime din spațiul pe care îl ocupă un *m*. Exemplele de fonturi monospace includ Courier sau American Typewriter; ele sunt utilizate spre a reprezenta linii de cod sau lucruri pe care utilizatorul ar trebui să le tasteze.

OBSERVAȚIE

De obicei se recomandă alegerea fonturilor serif pentru textul de rând și a fonturilor sans serif pentru titluri sau blocuri mici de text.

Există un motiv serios pentru această opțiune. Cercetarea și experiența arată că fonturile serif sunt mai ușor de citit. Cu toate acestea, fonturile sans serif devin tot mai populare ca fonturi pentru textul de rând în browserele Web, parțial întrucât cele serif nu se transpun întotdeauna corect peste o varietate de ecrane și rezoluții de ecran. Caracterele sans serif nu posedă ornamentația suplimentară, care creează probleme. Totul este o chestiune de apreciere.

Este necesar să vă întemeiați utilizarea fonturilor serif ori sans serif pe considerația că paginile trebuie să fie atractive și ușor de citit.

SUGESTIE

Utilizați o familie de caractere tipografice ca ultim articol într-o listă `font-family`, deoarece ea cuprinde opțiunile dumneavoastră asupra fontului cât mai complet posibil. Chiar dacă un caracter tipografic nu este disponibil pe un calculator dat, este probabil să fie disponibil unul asemănător, din aceeași categorie. Un designer priceput include o listă de variante, de la cea mai potrivită până la ultima, și sfârșește cu categoria. De exemplu, dacă preferați trei tipuri de caractere serif, atunci atributul final apare ca mai jos:

```
{font-family: garamond, goudy, times, serif}
```

■ **`font-size`**: Dimensionarea în foile de stil oferă designerului cinci opțiuni pentru măsurare:

puncte Pentru a preciza dimensiunea unui font în puncte, utilizați abrevierea `pt` imediat după valoarea numerică: `{font-size: 12pt}`. Acest standard este cel pe care îl utilizează cei mai mulți designeri.

pixeli Pixelii sunt specificați cu abrevierea `px`: `{font-size: 24px}`. Un pixel este un punct pe un ecran; numărul de pixeli variază cu mărimea ecranului. Mulți utilizează un ecran de mărime 800x600 pixeli, deși există încă o mulțime de ecrane de 640x480. Pentru imaginile de ecran capturate pentru această carte s-a folosit varianta 640x480.

SUGESTIE

Cei mai mulți designeri de aplicații Web sunt mai acomodați cu valorile în puncte sau pixeli pentru stabilirea dimensiunii fontului. Totuși, dacă preferați o altă metodă și o considerați ușor de utilizat pentru dumneavoastră, este la fel de bine. Cel mai bun sfat pe care îl pot da este de a alege o metodă și a o aplica. Consecvența este foarte importantă la crearea de stiluri și face mai ușoară adaptarea stilului pentru a se potrivi cu cerințele în schimbare.

inci Dacă v-ați stabilit fonturile în inci, pur și simplu folosiți `in` și valoarea numerică a dimensiunii fontului de care aveți nevoie, în inci: `{font-size: 1in}`.

centimetri Unii preferă centimetri, reprezentați prin `cm` și folosiți în același mod ca punctele și incii: `{font-size: 5cm}`.

procente Ca un ultim mijloc, ați putea dori să stabiliți un procentaj din dimensiunea în puncte: `{font-size: 150%}`.

■ **`font-style`**: Acest atribut impune, de obicei, stilul textului, de exemplu plasarea sa în caractere cursive. Iată sintaxa potrivită pentru a face aceasta:

```
{font-style: italic}.
```

Ați putea crede că o altă valoare pentru `font-style` ar putea fi bold (îngroșat), dar aceasta este cuprinsă în `font-weight`. Celelalte valori permise pentru acest atribut sunt `normal` și `oblique`; `normal` convertește textul la starea sa prestabilită, iar `oblique` este similar lui `italic` exceptând faptul că este înclinat manual, în loc de a se bizui pe adevărata versiune cursivă a respectivului aspect de literă.

■ **`font-weight`**: Consistența unui stil de caractere este numită *grosimea* sa. Ca și în cazul aspectului fontului, grosimile se bazează pe existența fontului și grosimii corespunzătoare pe mașina unei persoane. În foile de stil, este disponibil un interval de valori, incluzând următoarele: `extra-light`, `demi-light`, `light`, `medium`, `extra-bold`, `demi-bold` și `bold`.

Înainte să fixați grosimile de font, asigurați-vă că aspectul fontului căruia îi aplicați grosimea are disponibilă această grosime. Verificați-vă întotdeauna munca pe o varietate de platforme și mașini, când este posibil, spre a vedea dacă ați fost capabil să obțineți un proiect robust, chiar dacă s-ar putea ca unele mașini să nu accepte fontul sau grosimea fontului în cauză. Ca regulă generală, `bold` este totdeauna disponibil.

■ **`text-decoration`**: Acest atribut decorează textul; el include valorile `none`, `underline`, `italic` și `line-through`.

OBSERVAȚIE

Atributul `text-decoration` este preferatul persoanelor cărora le displac legăturile subliniate. Puteți folosi definiția de stil `A {text-decoration: none}` spre a întrerupe total hiperlegăturile subliniate. Pentru stilul inline, plasați valoarea în interiorul legăturii pe care doriți să o controlați:

```
<a style="text-decoration: none" href="nowhere.html" alt="Nowhere">
Link to nowhere</a>
```

■ **`line-height`**: Un alt aspect important legat de text este *interlinierea* (leading), care înseamnă mărimea spațiului dintre liniile textului. Acest spațiu ar trebui să aibă peste tot aceeași valoare, altfel rezultatul este o spațiere neregulată, neatractivă. Puteți folosi atributul `line-height` spre a stabili distanța dintre liniile de bază, adică părțile de jos ale rândurilor textului.

Pentru a stabili interlinierea unui paragraf, folosiți atributul `line-height` în puncte, inci, centimetri, pixeli sau procentaj, în același mod în care descrieți atributele `font-size`:

```
P {line-height: 14pt}
```

SUGESTIE

Când folosiți interlinierea cu atributul `line-height`, poate apărea un spațiu mare între titlurile dumneavoastră și textul care urmează. Această spațiere este posibilă deoarece interlinierea acționează în foile de stil cu referire la linia de bază a unei părți de text. Pentru a evita apariția unui spațiu suplimentar între titlu și corpul de text, stabiliți marginile de sus ale paragrafului la 0. Va exista în continuare un spațiu, dar acesta va fi de dimensiune firească față de echivalentul interlinie. Un alt truc este de a vă asigura că interlinierea este mai mică decât dimensiunea în puncte a textului de rând.

În secțiunea următoare, veți lucra asupra marginilor, indentărilor și a altor atribute de aliniere a textului. Ele vă vor ajuta să adăugați spațiu alb și puțină textură paginilor dumneavoastră.

Margini, indentări și atribute de aliniere a textului

În ciuda a ceea ce v-ar fi lăsat să credeți versiunea standard HTML, paginile nu există doar ca să aveți unde plasa text din margine în margine cu un minim de spațiu alb. Crearea unui proiect bun include stabilirea marginilor și a indentărilor astfel încât ochiul cititorului să fie condus de la un element la următorul și, având suficient spațiu alb, să acorde cititorului un repaus între reflecții. Următorul set de atribute servește acestui nobil scop:

■ **`margin-left`**: Pentru a stabili marginea la stânga, folosiți o distanță în puncte tipografice, inci, centimetri sau pixeli. Ceea ce urmează stabilește marginea la stânga la trei sferturi de inci:

```
{margin-left: .75in}
```

■ **`margin-right`**: Pentru marginea la dreapta, selectați una dintre opțiunile referitoare la măsură prevăzute pentru atributul `margin-left`:

```
{margin-right: 50px}
```

■ **margin-top:** Stabiliți marginile superioare utilizând aceleași valori de măsură ca și pentru alte atribute de margine:

```
{margin-top: 20pt}
```

■ **text-indent:** Din nou, valorile în puncte tipografice, inci, centimetri sau pixeli pot fi alocate acestui atribut, care indentează orice fel de text:

```
{text-indent: 0.5in}
```

Internet Explorer acceptă și valori negative ale marginii sau începutului de paragraf, pe care le puteți utiliza spre a crea efecte interesante și neobișnuite, cum ar fi text suprapus și capete de paragraf agățate (outdentate), pentru un design contemporan.

■ **text-align:** Cu această caracteristică îndelung așteptată, puteți alinia textul. Valorile pot fi `left`, `center` și `right`.

```
{text-align: right }
```

Alinierea textului este un instrument puternic de aranjare și, ca designer, vă veți bucura să fiți capabil să plasați text într-o varietate de aranjamente fără a trebui să vă bazați pe tabele, diviziuni sau alte soluții HTML ocolitoare, mai puțin elegante, din trecut. Nu uitați, alinierea textului cere un ochi fin. Alinierea la stânga este singura alegere rezonabilă pentru blocuri lungi de text, deoarece este mai ușor de citit și de urmărit pentru ochi. Alinierea la dreapta vine tocmai bine pentru scurte apariții de text, cum ar fi citatele. Folosiți textul centrat cu moderație. Chiar dacă pare natural să doriți să centrați textul, în realitate este mai dificil de citit și arată stereotip și greoi.

SUGESTIE

Alinierea completă (la dreapta și la stânga în același timp) nu este, încă, disponibilă. Singura soluție ocolitoare la îndemână este de a crea textul ca imagine grafică, iar apoi de a încărca grafica acolo unde doriți text aliniat. Este un mod greoi de realizare a sarcinii, dar alinierea completă ar putea fi adăugată ca opțiune într-o viitoare versiune a HTML. În acest moment, asigurați-vă că paginile dumneavoastră Web folosesc margini generoase și spațiu alb spre a conduce privirea cititorului acolo unde doriți dumneavoastră să meargă.

Cu aceste atribute în mână, este timpul să lucrăm pe un alt exemplu spre a vedea cum acționează ele într-o pagină HTML.

Controale pentru margini și aliniere

Pentru a controla marginile într-o foaie de stil, începeți cu o pagină HTML simplă care include o foaie de stil de bază, cum ar fi aceasta:

```
<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 4.0//EN">
<html>

<head>
<title>Styles</title>
</head>

<style>
<!--
BODY {background: white}
H1 {font-family: verdana, helvetica, arial, san-serif ;
➥font-size: 18pt; font-style: normal}
H2 {font-family: verdana, helvetica, arial, san-serif ;
➥font-size: 14pt; font-style: normal}
P {font-family: garamond, goudy, times, serif;
➥font-size: 12pt; font-style: normal; line-height: 14pt}
```



```
A {text-decoration: none; font-weight: bold}
-->
</style>

<BODY>
<H1>The Destination</H1>

<P>Nevertheless, in this <A href="mansion.html">mansion of gloom</A>
I now proposed to myself a sojourn of some weeks. Its proprietor,
Roderick Usher, had been one of my boon companions in boyhood; but
many years had elapsed since our last meeting.</P>

<H2>Why I came</H2>

<P>A <A href="letter.html">letter</A>, however, had lately reached me
in a distant party of the country--a letter from him--which, in its
wildly importunate nature, had admitted of no other than a personal
reply. The MS gave evidence of nervous agitation.</P>

</body>

</html>
```

Adăugați următoarea sintaxă a marginii pentru atributul `BODY` din secțiunea etichetei `<STYLE>`:

```
BODY {margin-left: 0.75in; margin-right: 0.75in; margin-top: 0.25in;
➥background: white }
```

Utilizați stilul `BODY` ca bază pentru alte elemente. Spre exemplu, alegerea stilului de text `Garamond` pentru corpul principal face din Garamond stilul de text de bază pentru paragrafe, hiperlegături și alte articole.

Salvați fișierul sub numele `margin.html` și vizualizați-l cu un browser. Rezultatele în Internet Explorer sunt arătate în Figura 18.6.

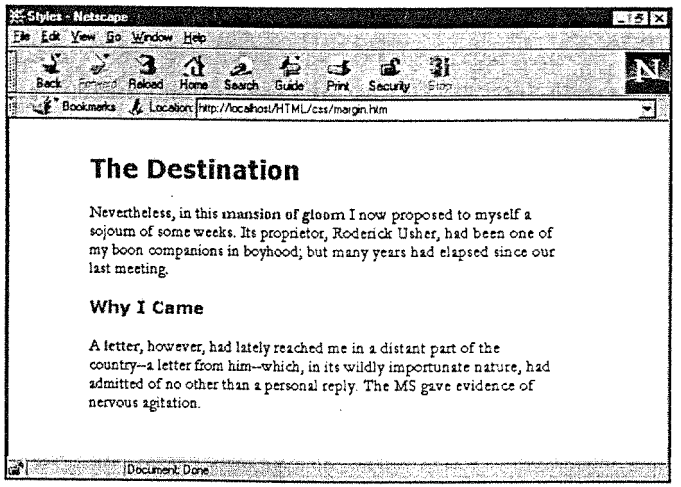


FIGURA 18.6. *Marginile creează spații albe atractive în jurul paginii, oferindu-i un „spațiu de respirație" față de marginea ferestrei browserului.*

În acest exemplu, valorile marginii pentru întreaga pagină au fost schimbate folosind atributul `BODY`. De asemenea, puteți adăuga margini oricărei alte etichete HTML. De exemplu, dacă doriți să stabiliți margini diferite pentru anteturi, plasați valorile pentru margine în șirul următor antetului, după alegerea dumneavoastră. În mod similar, puteți

ajusta marginile anumitor paragrafe adaugând șirului de paragraf valorile dorite pentru margini.

Dacă includeți margini pentru definiția de stil BODY pe lângă cele din definiția de stil P, asigurați-vă că marginile paragrafului sunt mai cuprinzătoare decât acelea pe care le-ați selectat pentru corpul textului. Altfel, nu veți fi capabili să vedeți diferența. Browserul ignoră marginea inferioară și folosește cea mai mare valoare disponibă dintre cele două.

După aceea, adăugați aliniere spre a creea un mai mare interes vizual pe pagină. Alinierea –*justification* – joacă un rol important în proiectare. Exemplul care urmează ilustrează modul în care schimbarea alinierii influențează textul.

Pornind de la exemplul `margin.html`, adăugați atributul `text-align` celor două etichete de titlu:

```
BODY {margin-left: 0.75in; margin-right: 0.75in; margin-top: 0.25in;
➥background: white}

H1 {font-family: verdana, helvetica, arial, san-serif;
➥font-size:18pt; font-style:normal; text-align:left}
H2 {font-family: verdana, helvetica, arial, san-serif ;
➥font-size:14pt; font-style:normal; text-align:right}
P {font-family: garamond; font-size: 12pt; font-style: normal;
➥line-height: 11pt}
A {text-decoration: none; font-weight: bold}
```

Întrucât aliniați paragrafe separat, adăugați acele atribute cu stilul inline. Întregul fișier ar trebui să arate astfel:

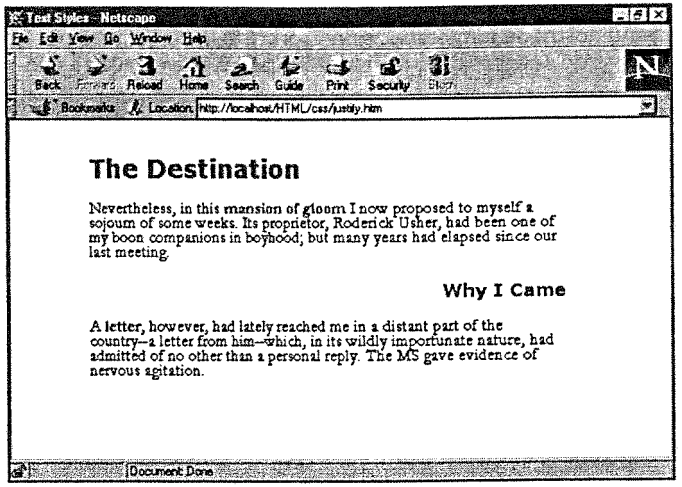
```
<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 4.0//EN">
<HTML>

<HEAD>
<TITLE>Text Styles</TITLE>
</HEAD>

<STYLE>
<!--
BODY {margin-left: 0.75in; margin-right: 0.75in; margin-top: 0.25in;
➥background: white}
H1 {font-family: verdana, helvetica, arial, san-serif ;
➥font-size:18pt; font-style:normal; text-align:left}
H2 {font-family: verdana, helvetica, arial, san-serif ;
➥font-size:14pt; font-style:normal; text-align:right}
P {font-family: garamond; font-size: 12pt; font-style: normal;
➥line-height: 11pt}
A {text-decoration: none; font-weight: bold}
-->
</STYLE>

<BODY>
<H1>The Destination</H1>

<P>Nevertheless, in this <A href="mansion.html">mansion of gloom</A>
I now proposed to myself a sojourn of some weeks. Its proprietor,
Roderick Usher, had been one of my boon companions in boyhood; but
many years had elapsed since our last meeting.</P>

<H2>Why I came</H2>

<P>A <A href="letter.html">letter</A>, however, had lately reached me
in a distant party of the country--a letter from him--which, in its
```



```
wildly importunate nature, had admitted of no other than a personal
reply. The MS gave evidence of nervous agitation.</P>

</BODY>

</HTML>
```

Salvați această pagină sub numele `justify.html` și vizualizați-o cu browserul dumneavoastră. Aspectul paginii devine mai interesant, așa cum este arătat în Figura 18.7.

FIGURA 18.7. *Alinierea ajută să facem o pagină Web mai interesantă vizual, chiar fără imagini.*



Până acum, ați învățat cum să schimbați aspectul de literă și atributele sale asociate, să schimbați stilul acesteia și să adăugați spațiere și aliniere în cadrul paginii. Acum, este timpul să scoateți o pensulă și să vedem ce se poate întâmpla cu culoarea.

Culoare și fundal

Aruncați o privire definiției de stil BODY și veți observa un atribut `background`, folosit spre a stabili culoarea de fundal a întregii pagini. De asemenea, puteți folosi atributul `bgcolor` cu eticheta <BODY>, deși utilizarea sa nu este recomandată în HTML 4.0. Atributul de stil `background` nu a fost dintotdeauna folosit în implementările timpurii ale CSS, iar `bgcolor` a fost utilizat ca o soluție ocolitoare.

SUGESTIE

Acum, când CSS este o parte stabilă atât pentru Navigator, cât și pentru Internet Explorer, puteți renunța la folosirea atributului `bgcolor`.

Puteți adăuga culoare de fundal altor stiluri, nu numai celui pentru BODY. De exemplu, puteți plasa o pată de culoare în spatele unui paragraf sau al unui antet pur și simplu introducând atributul `background` în eticheta respectivă. În plus, puteți schimba culoarea textului pentru orice etichetă, ceea ce este satisfăcător în special pentru designerii care caută în permanență să utilizeze culori bazate pe browser spre a însufleți paginile, în loc să se bizuiască pe soluții grafice, mari consumatoare de timp.

Sintaxa cerută spre a crea culoare de fundal este convenția atributului de stil `background` și o valoare hexazecimală pentru culoare:

```
{background: #FFFFFF}
```

Valoarea pentru culoare este un triplet hexazecimal, care este convenția Web standard pentru definirea unei culori. Începeți cu roșu, verde și albastru (cele trei culori de bază pentru un sistem aditiv). Dacă stabiliți valoarea maximă pentru toate cele trei împreună, obțineți alb. Îndepărtați-le pe toate trei și obțineți negru. Măriți-le sau micșorați-le pe toate trei cu aceeași valoare și obțineți tonuri de gri.

După ce ați terminat, fiecare pereche de digiți din valoare reprezintă una dintre culori – în ordine, roșu, verde și albastru. Cei doi digiți sunt un număr hexazecimal reprezentat în sistemul nostru de numerație zecimal prin 0 (00 hexazecimal) până la 255 (FF hexazecimal). Astfel, verde se reprezintă prin `00FF00`, galben este `FFFF00` ș.a.m.d.

OBSERVAȚIE

Internet Explorer permite nume de culori pentru atributele `color` și `background`. Numele de culori sunt: black (negru), silver (argintiu), gray (gri), white (alb), maroon (castaniu), red (roșu), purple (purpuriu), fuchsia (roșu-purpuriu), green (verde), lime (vernil), olive (măsliniu), yellow (galben), navy (bleumarin), blue (albastru), teal (albastru-verzui) și aqua (turcoaz).

În mod asemănător, puteți apela grafica de fundal înlocuind argumentul `hex` cu o adresă URL:

```
{background: http://myserver.com/cool.gif}
```

Schimbați culoarea textului utilizând atributul `color` și o valoare sau un nume pentru culoare:

```
{color: #FF6633}
```

Când folosiți `color` și `background`, asigurați-vă că lucrați cu combinații compatibile sau veți sfârși într-o harabaură de necitit. De exemplu, textul roșu pe un fond albastru este aproape imposibil de citit, dar galben pe fond albastru lucrează foarte bine.

Adăugarea culorii

În exemplul care urmează, veți adăuga o culoare de fundal unui titlu, lăsând restul paginii cu culoarea sa prestabilită.

Deschideți fișierul `justify.html` (folosit în exemplul anterior) în editorul dumneavoastră HTML. Adăugați `background:#99CCCC` variabilei de stil `H2`, astfel că aceasta devine:

```
H2 {font-family: verdana, helvetica, arial, san-serif; font-size:14pt;
➥font-style:normal; text-align:right; background: #99CCCC}
```

Salvați fișierul sub numele `color.html` și vizualizați-l cu browserul dumneavoastră. Rezultatul este arătat în Figura 18.8.

Contrastul pe care îl câștigați utilizând culori diferite poate fi foarte eficient, dar trebuie să fiți grijuliu și ferm în alegerea paletei de culori. Dacă este bine făcut, contrastul de culoare poate adăuga o fărâmă suplimentară de expresivitate și stil unei pagini Web.

Același procedeu funcționează și în cazul secțiunilor de text, urmărind aceleași linii directoare ca la `background` și utilizând atributul `color`:

1. Deschideți `color.html` în editorul dumneavoastră.
2. Adăugați `color:#FF0033` după eticheta `background` pentru stilul `H2`:
   ```
   H2 {font-family: verdana, helvetica, arial, san-serif;
   ➥font-size:14pt; font-style:normal; text-align:right;
   ➥background: #99CCCC; color: #FF0033}
   ```
3. Salvați fișierul sub numele `color2.html` și vizualizați-l cu un browser. Rezultatul este o pagină cu o bogăție de stil, deplin controlat prin utilizarea unei foi de stil înglobat.

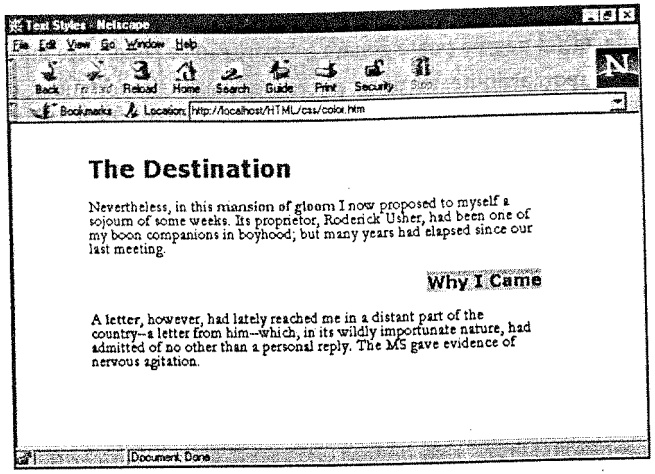


FIGURA 18.8. *Culoarea din spatele celui de al doilea titlu se evidențiază față de fundalul alb al ecranului.*

Cu unele prelucrări fundamentale de stil și înfățișare la cingătoare, este timpul să ne continuăm drumul către chestiuni de organizare mult mai lumești. Următoarea secțiune arată cum să definim foile de stil ca grupuri și să creăm subclase de diferite stiluri.

Organizarea stilului

Există două tehnici de organizare care ne pot ajuta să facem foile de stil mai ușor de utilizat. Prima este *gruparea*, care reduce volumul tastărilor necesare prin crearea grupurilor logice de stiluri. Dacă aveți nevoie de variații într-un stil, puteți folosi *clase*. Această tehnică vă permite să atribuiți stiluri diferite aceleiași etichete HTML.

Gruparea foilor de stil

Prima tehnică de grupare implică utilizarea acelorași atribute de stil pentru un set de etichete. Fără grupuri, sintaxa ar arăta ca mai jos:

```
H1 {font-family: arial; font-size: 14pt; color: #000000}
H2 {font-family: arial; font-size: 14pt; color: #000000}
H3 {font-family: arial; font-size: 14pt; color: #000000}
```

Iată același exemplu grupat:

```
H1, H2, H3 {font-family: arial; font-size: 14pt; color: #000000}
```

Suplimentar grupării de stiluri, puteți grupa atribute în interiorul unui stil utilizând un șir de valori într-o categorie de atribut. Utilizând sintaxa din exemplele anterioare, iată modul lung de scriere a unui stil pentru `BODY`:

```
BODY {font-family: arial, san-serif; font-size: 12pt; line-height: 14pt;
➥font-weight:bold; font-style: normal}
```

Prin gruparea de atribute, toate variabilele sunt reunite sub `font`. Rezultatul arată cam așa:

```
BODY {font: bold 12pt/14pt arial, san-serif}
```

Același procedeu se aplică altor grupuri de atribute. De exemplu, utilizați `margin` urmat de valorile pentru marginea de sus, din dreapta și din stânga:

```
BODY {margin: .10in .75in .75in}
```

Dacă folosiți două variabile, atunci al doilea număr se aplică atât marginii din dreapta, cât și celei din stânga. Dacă se folosește un singur număr, atunci acesta se aplică tuturor celor trei margini.

Luați aminte că variabilele sunt separate doar de un spațiu. Aveți grijă să nu folosiți virgule, altfel pagina poate genera o eroare sau se poate afișa într-un mod imprevizibil.

Când grupați atribute, nu uitați că ordinea lor este importantă. Grosimea fontului și stilul trebuie să vină înaintea altor atribute de stil; dimensiunea fontului vine înaintea interlinierii; abia apoi puteți adăuga informații suplimentare șirului. Observați că nu există vreo virgulă între atribute, exceptând cazul familiilor de fonturi. Mai multe informații despre ordinea corectă se regăsesc în Anexa C.

Atribuirea de clase

Puteți modifica un stil și prin atribuirea de clase etichetelor HTML individuale. Faceți aceasta adăugând un nume de extensie oricărei etichete HTML. Acest nume poate fi aproape orice, precum `BobText`, `hippo`, `topText`, `footerStuff` sau orice altceva care vă vine în minte. În exemplele precedente, există două stiluri pentru paragrafe – unul aliniază la stânga, iar celălalt la dreapta. În exemplu, ați utilizat stiluri inline spre a obține efectul pe care l-ați dorit, dar, utilizând clasele, îl puteți defini în avans:

```
P.left {font-family:garamond; font-size:12pt; font-style:normal;
➥line-height:11pt; text-align:left}

P.right {font-family:garamond; font-size:12pt; font-style:normal;
➥line-height:11pt; text-align:right }
```

În prima definiție, se creează un nou stil `<P>`, numit `left`. Acest stil este utilizat în document ca `<P class="left">`. El folosește text Garamond de 12 puncte care este aliniat la marginea din stânga. A doua definiție creează un nou stil `<P>`, numit `right`. Este utilizat ca `<P class="right">` și seamănă cu definiția de stil anterioară, exceptând faptul că aliniază textul la marginea din dreapta.

SUGESTIE

Este vremea potrivită de a utiliza ambele feluri de grupare spre a face mai simple crearea și editarea ulterioară a stilurilor. Începeți cu definirea etichetei `<P>`:

```
P {font:garamond 12pt/11pt normal}
```

După ce a fost definit terenul comun, puteți adăuga descrierile de clasă pentru cele două variații:

```
P.left {text-align:left}
P.right {text-align:right}
```

Când se pune la un loc, pachetul întreg arată astfel:

```
P {font:garamond 12pt/11pt normal}
P.left {text-align:left}
P.right {text-align:right}
```

Când vă întoarceți să revedeți stilul pentru schimbări sau depanări ulterioare, puteți vedea unde clasele au trăsături comune și unde sunt ele diferite. Este mult mai limpede și mult mai ușor de citit decât metoda originală.

Când utilizați stilul în document, utilizați atributul `class` spre a identifica versiunea etichetei care se aplică:

```
<P class="left">His reserve had been always excessive and habitual.</P>
<P class="right">His reserve had been always excessive and habitual.</P>
```

Toate etichetele `<P>` care includ `class="left"` folosesc atributele de stil `P.left`, iar etichetele `<P>` care includ `class="right"` folosesc atributele `P.right` definite pentru acea clasă.

Dacă folosiți noile definiții cu grupare pentru fișierul `justify.html`, secțiunea stil arată acum astfel:


```
<STYLE>
<!--
BODY {margin-left: 0.75in; margin-right: 0.75in; margin-top: 0.10in}

H1,H2 {font-family: verdana, helvetica, arial, san-serif}
H1 {font-size:18pt; text-align:left}
H2 font-size:14pt; text-align:right}

P {font:garamond 12pt/11pt normal}
P.left {text-align:left}
P.right {text-align:right}

A {text-decoration: none; font-weight: bold}
-->
</STYLE>
```

Fișierul completat se salvează sub numele `group.html`, cu fiecare linie de stil mult mai scurtă și succintă, ușor de citit și de înțeles. Rezultatul vizual este același (vezi Figura 18.9.).

`Group.html` folosește aceleași stiluri ca și `justify.html`, dar stilurile sunt definite într-un mod diferit. Deși definițiile utilizează o sintaxă diferită, nu veți observa nici o diferență între imaginea din Figura 18.9 și cea din Figura 18.7.

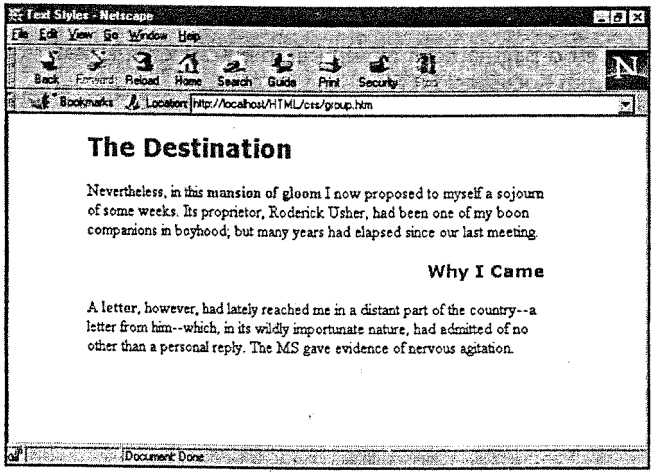


FIGURA 18.9. *Folosirea grupării de stil și de atribut creează aceeași pagină ca și* `justify.html`.

Pretutindeni foi de stil: o culegere de exemple

Deoarece foile de stil sunt atât de flexibile, acum este vremea să petrecem câteva clipe lucrând cu o diversitate de șabloane spre a ne familiariza cu tipurile de efecte posibile. Veți observa că majoritatea exemplelor nu implică foi de stil cu mari complicații. Opinia mea personală este: cu cât mai simplu, cu atât mai ușor de înțeles și de întreținut.

Exemplele care urmează schimbă numai câteva atribute ale stilurilor date, precum fonturi, culori și aliniere. Seria etichetelor afectate este, de asemenea, limitată și include stiluri de anteturi și paragraf. Totuși, ele sunt fundamentul foilor de stil pe care le puteți modifica și utiliza în propriile dumneavoastră pagini Web.

Toate exemplele utilizează foi de stil înglobat. Ca o caracteristică în plus, se dă și codul exemplului, asfel încât puteți vedea cum ar apărea ca stil legat, ceea ce este util pentru

folosirea aceleiași foi de stil în pagini multiple. De asemenea, luați aminte că foile de stil nu sunt singura metodă folosită spre a controla aspectul paginii. Sunt de asemenea utilizate pe larg tabelele, etichetele HTML și atributele.

SUGESTIE

Este necesar să fie spus o dată și încă o dată și încă o dată: trebuie să vă cunoașteți vizitatorii. Este mult mai important să transmiteți mesajul dumneavoastră decât să dovediți cât de incredibil de dibaci puteți fi cu foile de stil, în special când lucrați cu clienți din lumea afacerilor. Dacă este de ales între interesele de afaceri ale clientului și judecata cea mai bună a unui designer, alegeți interesul afacerii. Rentează mai mult pentru oricare dintre cei implicați.

Cu aceste exemple sper să vă trezesc interesul pentru foile de stil și să vă determin să vă rezervați ceva timp pentru a lucra cu ele. Este un bun proiect pentru seară, când gândul de a viziona reluările de la televizor vă face stomacul ghem. Cunoașterea foilor de stil vă va conferi putere asupra paginilor și vă va determina să lucrați cu Dynamic HTML cu mai multă plăcere.

Exemplul 1: O pagină de prezentare pentru un raport

Porniți cu o pagină simplă de prezentare pentru un raport online. Când veți începe să lucrați cu layere (straturi), mai târziu, în această săptămână, o veți putea folosi drept layer superior, cu alte layere sub ea.

Deschideți o pagină nouă în editorul dumneavoastră HTML și începeți cu structura de bază a unui document HTML.

```
<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 4.0//EN">
<HTML>

<HEAD>
<TITLE>Knee Deep in the Green Stuff</TITLE>
</HEAD>

<STYLE>
<!--

-->
</STYLE>

<BODY>
</BODY>

</HTML>
```

Observați etichetele de comentariu folosite între etichetele `<STYLE>` pentru a vă asigura că browserele incompatibile cu stiluri nu vor afișa cuprinsul ca alt text HTML.

Deoarece aceasta este o pagină de prezentare, păstrați-o simplă. Nu are rost să aglomerați ecranul cu grafică sau text de prisos. Folosiți un titlu, un subtitlu și numele organizației împreună cu o imagine. Adăugați toate aceste articole corpului paginii:

```
<BODY bgcolor=white>
<H1>Knee Deep</H1>
<IMG SRC="/mrfd/images/cleanup-r.gif">
<H2>Incident Analysis:<BR>Post Creek Poison Spill</H2>
<H3>Missoula Regional<BR>Hazardous Materials</H3>
</BODY>
```

Ca o opțiune prestabilită pentru toate exemplele, aduceți la alb culoarea de fundal a paginii.

SUGESTIE

Nu uitați, puteți folosi și metoda tripletului hexazecimal de specificare a culorilor. Tocmai am folosit cuvântul cheie `color` spre a-l face mai ușor de citit. Valoarea hexazecimală pentru alb este `#FFFFFF`.

Ați luat un start bun, chiar dacă nu e prea spectaculos, precum se vede în Figura 18.10. Ea arată cum apare pagina într-un browser fără stiluri. Este treaba dumneavoastră să faceți ceva mai mult din ea.

Tratamentul acestei pagini începe prin adăugarea unui chenar alb în jurul ei:

```
BODY {margin:.25in .5in}
```

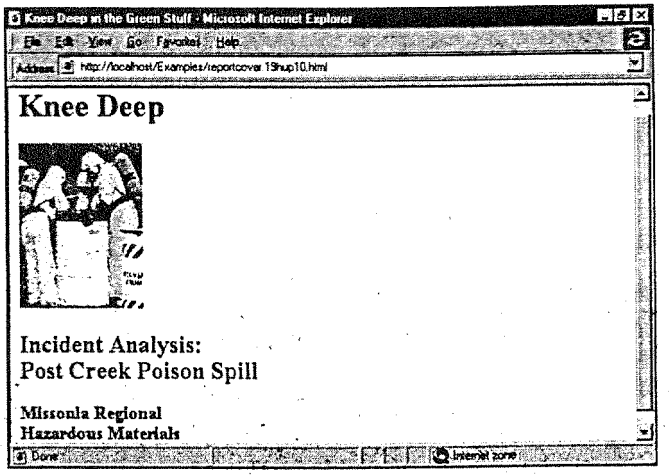


FIGURA 18.10. *Pagina de titlu inițială este utilă, dar nu are un design foarte interesant.*

Această definiție adaugă un sfert de inci de spațiu în partea de sus și în cea de jos a paginii și o jumătate de inci la stânga și la dreapta ei. Ea folosește gruparea de atribute spre a elimina necesitatea de a utiliza `margin-top`, `margin-bottom`, `margin-left` și `margin-right`, care implică mult mai multă tastare.

Apoi, stabiliți tipul de caracter pentru primele trei titluri:

```
H1, H2, H3 {font-family:helvetica, arial, sans-serif}
```

Deoarece aceasta este prima pagină a raportului, presupuneți că restul paginilor vor fi adăugate prin alipire la aceeași pagină. Puteți să presupuneți și că restul documentului nu necesită titluri aliniate la dreapta. Pentru a utiliza alinierea la dreapta doar în prima pagină, creați o nouă clasă, numită `page1` cu alinierea dorită pentru copertă:

```
H1.page1, H2.page1, H3.page1 {text-align:right}
```

Acum, încheiați lucrul asupra celor trei titluri prin stabilirea unor margini potrivite cu imaginea dumneavoastră, astfel încât coperta să nu depășească partea de jos a paginii:

```
H1.page1 {margin-bottom:-.15in}
H2.page1 {margin-top:-.15in; margin-bottom:0in}
H3.page1 {color:white; background-color:black; margin-top:0in}
```

Observați utilizarea numerelor negative. În loc să adauge spațiu liber, ele extrag spațiu, astfel că orice apare după `H1.page1` și înainte de `H2.page1` se poate apropia cu .15 inci de respectivele elemente.

Pentru a scoate în evidență secțiunea de pagină marcată cu <H3>, textul său este inversat la alb pe un fundal negru. Secțiunea de stil finisată arată acum astfel:

```
<STYLE>
<!--
BODY {margin:.25in .5in}

H1, H2, H3 {font-family:helvetica, arial, sans-serif}
H1.page1, H2.page1, H3.page1 {text-align:right}

H1.page1 {margin-bottom:-.15in}
H2.page1 {margin-top:-.15in; margin-bottom:0in}
H3.page1 {color:white; background-color:black; margin-top:0in}
-->
</STYLE>
```

Pentru a termina treaba, este nevoie să efectuați și alte modificări în corpul de pagină. Pentru aceasta, este nevoie să adăugați doar un atribut `class` fiecăreia dintre etichetele de titlu:

```
<H1 class="page1">Knee Deep</H1>
<IMG SRC="/mrfd/images/cleanup-r.gif">
<H2 class="page1">Incident Analysis:<BR>Post Creek Poison Spill</H2>
<H3 class="page1">Missoula Regional<BR>Hazardous Materials</H3>
```

Salvați fișierul sub numele `ReportCover.html`. El ar trebui să se potrivească cu Listingul 18.1.

Listingul 18.1. `ReportCover.html`.

```
<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 4.0//EN">
<HTML>

<HEAD>
<TITLE>Knee Deep in the Green Stuff</TITLE>
</HEAD>

<STYLE>
<!--
BODY {margin:.25in .5in}

H1, H2, H3 {font-family:helvetica, arial, sans-serif}
H1.page1, H2.page1, H3.page1 {text-align:right}

H1.page1 {margin-bottom:-.15in}
H2.page1 {margin-top:-.15in; margin-bottom:0in}
H3.page1 {color:white; background-color:black; margin-top:0in}
-->
</STYLE>

<BODY bgcolor=white>
<H1 class="page1">Knee Deep</H1>
<IMG SRC="/mrfd/images/cleanup-r.gif">
<H2 class="page1">Incident Analysis:<BR>Post Creek Poison Spill</H2>
<H3 class="page1">Missoula Regional<BR>Hazardous Materials</H3>
</BODY>

</HTML>
```

Când vizualizați lucrarea finală în Internet Explorer, ea arată ca în Figura 18.11. Această copertă poate fi refolosită în același scop câtă vreme imaginea rămâne apropiată de aceeași dimensiune. Dacă devine prea înaltă, atunci ea va forța textul să părăsească partea de jos a paginii și să ascundă o porțiune din proiect pe ecranele mai mici.

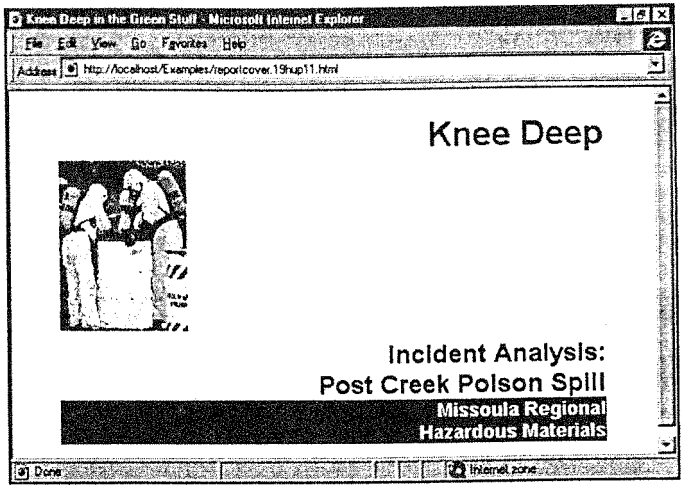


FIGURA 18.11. *Datorită elementelor de stil adăugate, pagina sugerează acum mai multă importanță.*

Pentru a realiza acest exemplu cu o foaie de stil legat, care să poată fi folosită și de către alte coperte de raport, decupați etichetele <STYLE> și conținutul lor și alipiți-le într-un fișier nou, denumit `ReportCover.css`. Acum, în antetul documentului original `ReportCover.html`, adăugați o linie nouă după eticheta de închidere </TITLE>:

```
<LINK rel=stylesheet href="ReportCover.css" type="text/css">
```

Va acționa exact la fel ca stilul înglobat, dar acum este ușor accesibilă de alte pagini care doresc să folosească același set de atribute.

Exemplul 2: Distracții pentru copii

Acest exemplu reprezintă o altă implementare de stil destul de simplă. Schimbările sunt puține, dar ele reușesc să învioreze o pagină adresată copiilor și părinților lor. Formatul este al unui buletin electronic, drept care accentul este pus pe conținut mai mult decât pe interactivitate.

Începeți cu documentul HTML de bază în editorul dumneavoastră (cum ați făcut în exemplul anterior). Buletinul este compus din patru părți fundamentale: frontispiciu (masthead), manșetă (banner), titluri (headlines) și text. Primele două apar o singură dată în fiecare pagină.

Frontispiciul este pus într-un font de titlu mai mic, manșeta este pusă în cel mai mare și ambele elemente sunt centrate în pagină:

```
<H5 align=center>Summer news and reviews * Where to find fun at Usher</H5>
<H1 align=center>KIDZ ZONE</H1>
```

Acum stabiliți titluri pentru fiecare poveste, cu conținutul egal împărțit pe trei coloane. Folosiți tabele pentru a conferi formatului pe trei coloane suficientă umplere ca textul să aibă „spațiu de respirat". O linie orizontală marchează sfârșitul unei povești:

```
<TABLE cellpadding="15">
<TR valign="top">
<TD>
<H2>A little about woodwork</H2>
In this there was much that reminded me of the specious totality of old
wood-work which has rotted for long years in some neglected vault, with no
```

```
disturbance from the breath of the external air. Beyond this indication of
extensive decay, however, the fabric gave little token of instability.
</TD>
<TD>
Perhaps the eye of a scrutinizing observer might have discovered a barely
perceptible fissure, which, extending from the roof of the building in
front, made its way down the wall in a zigzag direction, until it became
lost in the sullen waters of the tarn.
<HR>
<TD>
</TD>
<H2>The arrival</H2>
Noticing these things, I rode over a short causeway to the house. A
servant in waiting took my horse, and I entered the Gothic archway of the
hall. A valet, of stealthy step, thence conducted me, in silence, through
many dark and intricate passages in my progress to the studio of his
master.
<HR>
</TD>
</TR>
</TABLE>
```

Prin urmare, buletinul arată ca în Figura 18.12, ceea ce nu e rău – doar că nu e prea bine.

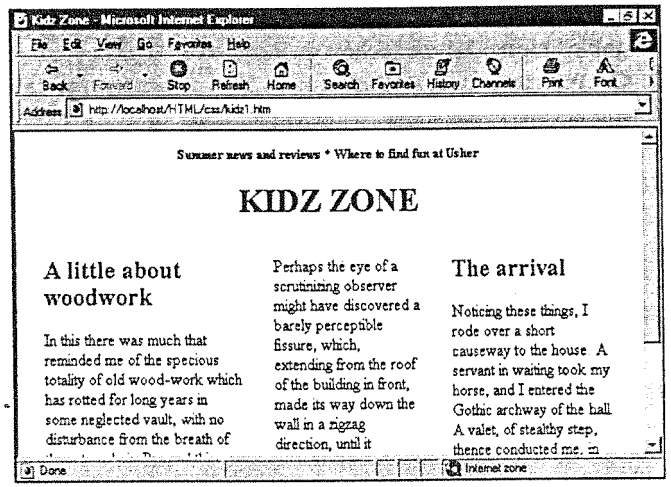


FIGURA 18.12. *Lansarea buletinului evidențiază nevoia de ajutor vizual, în primul rând în frontispiciu și manșetă.*

Acum lucrați la adăugarea unor condimente pentru a face pagina un pic mai interesantă. Întâi, adăugați ceva spațiu alb în părțile de sus și laterale pentru a oferi proiectului dumneavoastră puțină independență față de fereastra browserului. Apoi, continuați cu frontispiciul, manșeta și titlurile. Folosiți pentru toate trei Comic Sans, un tip de caracter ușor glumeț, dar totdeauna foarte lizibil.

```
BODY {margin-left:36pt; margin-right:36pt; margin-top:24pt }
H1, H2, H5 {font-family:"comic sans ms", verdana, sans-serif;
➥text-align:center}
```

Urmează detaliile pentru frontispiciu și manșetă. Așezați frontispiciul pe un fundal galben, iar manșeta în galben pe un fundal albastru:


```
H1 {font-size:60pt; color:yellow; background-color:blue;
➥margin-top:0pt; margin-bottom:18pt}
H5 {font-size:12pt; background-color:yellow;
➥margin-top:36pt; margin-bottom:0pt}
```

Introduceți un spațiu suplimentar între manșetă și tabelul de mai jos prin utilizarea atributului `margin-bottom`.

SUGESTIE

Probabil vă mirați de ce nu ați folosit albastru pe galben în frontispiciu, în opoziție cu galbenul pe fond albastru al manșetei. Aceasta deoarece albastru pe galben arată destul de apropiat de negru, iar caracterele negre sunt un pic mai ușor de citit. Deci ați lăsat textul frontispiciului cu negrul prestabilit și doar ați adăugat un fond galben.

Prelucrați titlurile asemănător frontispiciului și manșetei, dar aliniați-le la stânga, astfel încât să fie mai ușor de citit. De asemenea, oferiți liniilor orizontale de la sfârșitul fiecărei povești o ușoară tușă suplimentară prin colorarea lor în verde și faceți-le suficient de groase încât să fie observate. Verdele reprezintă un accent drăguț pentru albastrul cu galben din capătul paginii. Definiți o linie mare, neagră pentru linia orizontală normală, cu care să marcați sfârșitul paginii:

```
H2 {font-size:18pt; text-align:left}
HR {color:black; height:16}
HR.green {color:green; height:8}
```

Ultimul în listă este textul de rând. În loc de a folosi etichete de paragraf, exploatați avantajul etichetei pentru date în tabel. În unele cazuri, eticheta `<TD>` nu este folosită cu o etichetă de închidere. Pentru ca acest stil să funcționeze, eticheta `</TD>` este necesară:

```
TD {font-family: tahoma, arial, sans-serif; font-size:12pt; width:33%}
```

Atributul `width` de la sfârșit stabilește lățimea celulei de tabel, care este aceeași ca atunci când folosim `<TD width="33%">`. Definiția de stil finisată arată acum astfel:

```
<STYLE>
<!--
BODY {margin-left:36pt; margin-right:36pt; margin-top:24pt;
➥background:white}
H1, H2, H5 {font-family:"comic sans ms", verdana, sans-serif;
➥text-align:center}
H1 {font-size:60pt; color:yellow; background-color:blue;
➥margin-top:0pt; margin-bottom:18pt}
H2 {font-size:18pt; text-align:left}
H5 {font-size:12pt; background-color:yellow;
➥margin-top:36pt; margin-bottom:0pt}
TD {font-family: tahoma, arial, sans-serif; font-size:12pt; width:33%}
HR {color:black; height:16}
HR.green {color:green; height:8}
-->
</STYLE>
```

Tot ce rămâne de făcut este să adăugați numele clasei etichetelor de linie orizontală de la sfârșitul poveștilor:

```
<HR class="green">
```

Când ați terminat, salvați fișierul sub numele `Kidz.html`. Ar trebui să arate ca Listingul 18.2.

Listingul 18.2. Kidz.html.

```
<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 4.0//EN">
<HTML>

<HEAD>
<title>Kidz Zone</title>
</HEAD>

<STYLE>
<!--
BODY {margin-left:36pt; margin-right:36pt; margin-top:24pt;
➥background:white}
H1, H2, H5 {font-family:"comic sans ms", verdana, sans-serif;
➥text-align:center}
H1 {font-size:60pt; color:yellow; background-color:blue;
➥margin-top:0pt; margin-bottom:18pt}
H2 {font-size:18pt; text-align:left}
H5 {font-size:12pt; background-color:yellow;
➥margin-top:36pt; margin-bottom:0pt}
TD {font-family: tahoma, arial, sans-serif; font-size:12pt; width:33%}
HR {color:black; height:16}
HR.green {color:green; height:8}
-->
</STYLE>

<BODY>
<H5>Summer news and reviews * Where to find the fireworks</H5>
<H1>KIDZ ZONE</H1>
<TABLE cellpadding="15">
<TR valign="top">
<TD>
<H2>A little about woodwork</H2>
In this there was much that reminded me of the specious totality of old
wood-work which has rotted for long years in some neglected vault, with no
disturbance from the breath of the external air. Beyond this indication of
extensive decay, however, the fabric gave little token of instability.
</TD>
<TD>
Perhaps the eye of a scrutinizing observer might have discovered a barely
perceptible fissure, which, extending from the roof of the building in
front, made its way down the wall in a zigzag direction, until it became
lost in the sullen waters of the tarn.
<HR class="green">
</TD>
<TD>
<H2>The arrival</H2>
Noticing these things, I rode over a short causeway to the house. A
servant in waiting took my horse, and I entered the Gothic archway of the
hall. A valet, of stealthy step, thence conducted me, in silence, through
many dark and intricate passages in my progress to the studio of his
master.
<HR class="green">
</TD>
</TR></TABLE>
<HR>
</BODY>

</HTML>
```


Când vizualizați pagina cu Internet Explorer, ea arată ca în Figura 18.13. Este, într-adevăr, un stil simplu, dar culoarea suplimentară și dimensionarea caracterelor produc o mare diferență în impactul său, chiar fără utilizarea imaginilor.

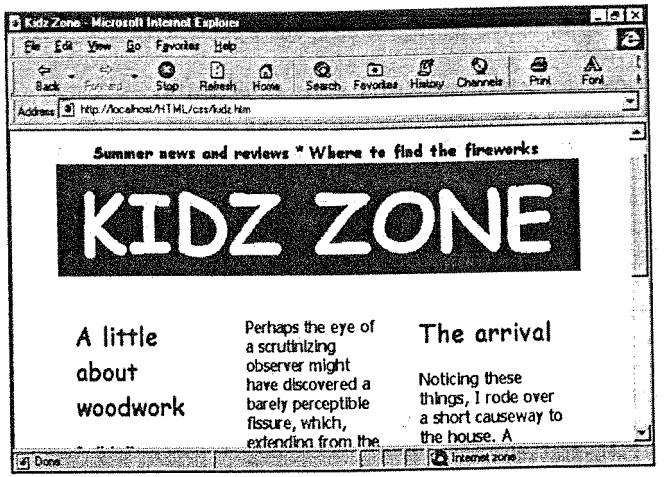


FIGURA 18.13. *Folosirea a două tipuri de caractere și a trei culori pentru a obține un efect cu mult mai multă atracție vizuală decât a modelului original.*

Exemplul 3: Variațiune asupra unei broșuri

Acest exemplu este denumit broșură, dar s-ar putea potrivi unei mulțimi de ținte într-un sit Web, precum un catalog, una dintr-o serie de întrebări frecvente (FAQ – Frequently Asked Questions) sau chiar o pagină de confirmare. În el se folosesc margini negative, la fel ca în primul exemplu.

Începeți cu o pagină de start în editorul dumneavoastră HTML (cum ați făcut în primul exemplu). Acest proiect se bazează pe un format cu două coloane, cu titlul pe stânga și textul pe dreapta. Doriți mult spațiu între cele două elemente, deci stabiliți umplerea la 20 de pixeli. Ba mai mult, această pagină ar trebui să arate la fel indiferent ce fel de monitor are utilizatorul, deci luați-vă și libertatea de a stabili lățimea coloanelor în pixeli:

```
<TABLE cellpadding="20">
<TR>
<TD width="150">
<H2>Wild</H2>
<H2>Flowers</H2>
</TD>
<TD width="300">
<P>I looked upon the scene before me--upon the mere house, and the simple
landscape features of the domain--upon the bleak walls--upon the vacant
eye-like windows--upon a few rank sedges--and upon a few white trunks of
decayed trees--with an utter depression of soul which I can compare to no
earthly sensation more properly than to the after-dream of the reveller
upon opium--the bitter lapse into everyday life--the hideous dropping off
of the veil.</P>
</TR>
</TABLE>
```

Primele sunt lucruri de rutină. Definiți o margine de un sfert de inci pentru partea de sus și partea de jos și o jumătate de inci pentru fiecare dintre părțile laterale. Apoi stabiliți stilul de caracter de bază pentru titlu și aplicați paginii un fundal alb (nu țin neapărat la acest detaliu):

```
BODY {margin:.25in .5in; background:white}
H2 {font-family:serif; font-size:14; text-align:center}
```

Pentru a obține efectul dorit, despărțiți eticheta de titlu în trei clase separate: `base`, `reverse` și `large`. `Base` este aceeași ca definiția `H2` de la ultimul pas, de aceea treceți la stilul `reverse`. Ar trebui să fie alb pe negru, cu spațiere suplimentară de literă spre a completa spațiul orizontal:

```
H2.reverse {letter-spacing:1.25em; color:white;
➥background-color:black}
```

Urmează `big class`. Ea include o literă inițială mare, de care vă puteți îngriji cu un stil inline puțin mai târziu. Pentru moment, măriți-i dimensiunea la 30 puncte, faceți-o cursivă (italică) și așezați-o forțat la marginea din dreapta. Marginea negativă de sus o obligă să se apropie de elementul care va apărea deasupra sa (`H2.reverse`):

```
H2.big {font-style:italic; font-size:30; text-align:right;
➥margin-top:-48pt}
```

Ultimul subiect este corpul de text al broșurii. Folosiți pentru el un font sans serif spre a contrasta cu titlul; adăugați spațiu suplimentar între linii și o spațiere ușor crescută între litere. Efectul amintește întrucâtva de semințele plantate într-o grădină de legume sau într-o grădină de flori foarte ordonată:

```
P {font:8pt verdana, tahoma, sans-serif; line-height:12pt;
➥letter-spacing:.1em}
```

Lăsați puțin stilurile deoparte și adăugați celor două titluri informații asupra clasei:

```
<H2 class="reverse">Wild</H2>
<H2 class="big">flowers</H2>
```

Bănuiți care va fi rezultatul? Cuvântul *Wild* va apărea în video invers într-o manșetă plasată deasupra cuvântului *flowers*, afișat cu caractere mult mai mari. Dar ce facem cu acea primă literă imensă? Citiți mai departe!

Folosiți eticheta `<SPAN>` pentru a marca prima literă. Această etichetă lucrează bine pentru a marca litere singulare sau blocuri întregi de text. Este o minunată etichetă de stil multifuncțională. Folosiți pentru litera mare o culoare diferită (roșu), pentru a se detașa de restul titlului și a adăuga o pată izbitoare de culoare:

```
<H2 class="big">
<span style="font-size:72pt; color:red; font-weight:normal">f</span>
lowers</H2>
```

Litera *f* a fost separată în această linie de restul cuvântului *flowers* pentru a se vedea mai ușor eticheta `<SPAN>`. Când pagina este afișată, litera *f* va apărea în linie cu restul cuvântului.

Salvați fișierul sub numele `DirtyFingernails.html` și afișați-l cu un browser. Dacă fișierul dumneavoastră este același ca în Listingul 18.3, atunci browserul ar trebui să dezvăluie ceva apropiat Figurii 18.14.

Listingul 18.3. `DirtyFingernails.html`.

```
<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 4.0//EN">
<HTML>

<HEAD>
<title>Dirty Fingernails -- Wildflowers</title>
</HEAD>

<STYLE>
<!--
```



```
BODY {margin:.25in .5in; background:white}
H2 {font-family:serif; font-size:14; text-align:center}
H2.reverse {letter-spacing:1.25em; color:white; background-color:black}
H2.big {font-style:italic; font-size:30; text-align:right; margin-top:-48pt}
P {font:8pt verdana, tahoma, sans-serif; line-height:12pt; letter-spacing:.1em}
-->
</STYLE>

<BODY>
<TABLE cellpadding="20">
<TR>
<TD width="150">
<P> </P>
<H2 class="reverse">Wild</H2>
<H2 class="big">
<span style="font-size:72pt; color:red; font-weight:normal">
f</span>lowers</H2>
</TD>
<TD width="300">
<P>I looked upon the scene before me--upon the mere house, and the simple
landscape features of the domain--upon the bleak walls--upon the vacant
eye-like windows--upon a few rank sedges--and upon a few white trunks of
decayed trees--with an utter depression of soul which I can compare to no
earthly sensation more properly than to the after-dream of the reveller
upon opium--the bitter lapse into everyday life--the hideous dropping off
of the veil.</P>
</TR>
</TABLE>
</BODY>

</HTML>
```

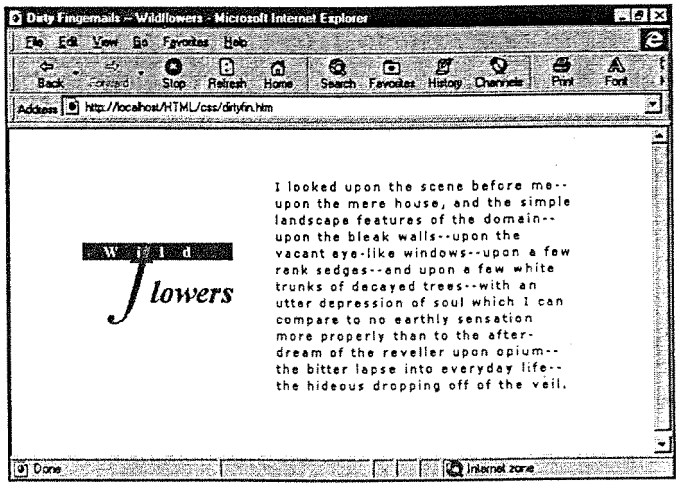


FIGURA 18.14. *Această broșură folosește o margine negativă și o literă mare pentru a crea efectul de text suprapus peste alt text.*

Neajunsul acestui exemplu este că, atunci când el apare într-un browser care este incompatibil cu foile de stil întregul efect este anulat. Totuși el apare într-o formă suficient de prezentabilă pentru a fi lizibilă și acceptabilă pentru utilizator.

Rezumat

Conceptul de foi de stil este foarte avansat pentru HTML. Ele vă oferă instrumente de proiectare pentru a face lucruri care nu au fost niciodată măcar gândite în zilele de început ale Webului. Totuși, implementarea prezentă este încă nerafinată și stângace. Ca și pentru majoritatea tehnologiilor asociate Internetului, nu vă așteptați ca situația să dureze mult. Datorită proceselor de colaborare din Consorțiul World Wide Web și competiției intense dintre Microsoft și Netscape, vă puteți aștepta la mai multe schimbări și progrese în lunile și anii care vor veni.

Dacă veți începe să folosiți foi de stil pentru a vă proiecta paginile sau pentru a vă finisa creațiile, în mod sigur vă veți lovi de limitările browserelor. Amintiți-vă, conceptul și tehnologia sunt noi, iar obstacolele vor cădea pe măsura înaintării în timp.

Capitolul 19
Foile de stil în cascadă și poziționarea elementelor

de Rick Darnell

În acest capitol

- **Poziționarea relativă contra poziționării absolute 261**
- **Alte proprietăți ale elementelor poziționate 268**
- **Combinarea poziționării și a scripturilor 270**

Unul dintre marile inconveniente ale utilizării limbajului HTML până în acest moment a fost lipsa de instrumente de așezare precisă în pagină. Această limitare a forțat autorii de pagini Web să recurgă la artificii cu tabele, cadre și imagini pentru a-și îndeplini scopurile de proiectare.

Cu standardul inițial pentru foi de stil în cascadă (vezi capitolul anterior, „Utilizarea foilor de stil în cascadă") au fost atinse multe dintre problemele de proiectare, incluzând margini, aliniere și dimensiunea textului. Totuși, cele mai multe dintre proprietățile CSS (Cascading Style Sheets) se referă la aspect, nu la poziționarea efectivă în pagină.

Prin intermediul Consorțiului World Wide Web (W3C), Netscape și Microsoft și-au reunit eforturile pentru a crea o anexă a standardului CSS care să prevadă un set nou de proprietați ale stilului doar pentru mutarea elementelor într-o pagină (`http://www.w3.org/TR/WD-positioning`) – și aceasta reprezintă ceea ce urmează să explorați în capitolul de față.

Poziționarea relativă contra poziționării absolute

Una dintre primele chestiuni de urmărit în poziționare este geografia: cum determinați unde să plasați un element în pagină? Plasarea necesită un sistem de coordonate cu puncte de start predictibile. În prezent, poziționarea CSS oferă două sisteme.

Primul este *relativ*. Ar trebui să îl recunoașteți, deoarece a fost sistemul de poziționare prestabilit de când a fost creat HTML. Folosind poziționarea relativă HTML tradițională, punctul de start al fiecărui element depinde de elementul anterior. De exemplu, priviți acest extras:

```
<H1>Here's the title</H1>
<P>Now is the time for <BR>someone to do something.</P>
```

Paragraful este așezat cu o linie mai jos pe pagină *relativ* la titlul de deasupra sa, iar punctul de start al textului după eticheta de salt la linie nouă este plasat *relativ* la linia precedentă. În cazul nostru, aceasta înseamnă plasarea sa pe linia următoare în același punct pe orizontală ca și începutul de paragraf.

Al doilea sistem de poziționare este *absolut*. Orice element care utilizează poziționarea absolută poate fi plasat oriunde pe pagină, independent de oricare alt element. Sistemul absolut începe numărarea la colțul din stânga-sus al documentului, așa cum este prezentat în fereastra browserului. Prin urmare, coordonatele colțului din stânga-sus sunt 0,0. Pe un display de 640x480 pixeli, colțul din dreapta-jos va avea coordonatele 620,440 sau undeva pe aproape.

OBSERVAȚIE

De ce nu este posibil să cunoaștem coordonatele exacte ale colțului din dreapta-jos? Din mai multe motive. Întâi, dimensiunea displayului utilizatorului nu poate fi determinată, aceasta putând fi la fel de bine 1024x768 sau 640x480. Apoi, dacă browserul este maximizat pe display, mai există un spațiu suplimentar necesar pentru bare de defilare, meniuri, bare de stare și alte caracteristici care consumă din spațiul disponibil. Sau browserul ar putea împărți ecranul cu alte aplicații; sau pagina însăși ar putea ocupa un cadru mai mic în interiorul ferestrei de browser curente.

În Windows 95/NT sau Microsoft Office, de exemplu, ați putea avea și o bară de taskuri sau o bară de unelte flotantă pe ecran, cu care browserul trebuie să se descurce. În sfârșit, browserul poate să nu fie maximizat.

Datorită naturii HTML și browserelor, nu există nici un mod riguros de a ști unde se află colțul din dreapta jos al displayului beneficiarului dumneavoastră. În consecință, toate coordonatele încep cu colțul din stânga-sus al displayului.

Cunoscând definiția de bază a sistemelor de poziționare relativă și absolută, rămâneți mirat de ce ar trebui să vă mai bateți capul cu poziționarea relativă. În definitiv, poziționarea absolută vă permite să puneți un element oriunde doriți pe pagină, și ar trebui să fie în același loc pe un același browser, cu aceeași rezoluție. Cu poziționarea relativă, poziția elementului tinde mereu să depindă de un alt element, ceea ce înseamnă că aveți un control mai redus.

Totuși, amintiți-vă că elementele care utilizează poziționarea absolută nu știu nimic unul de altul, astfel încât este mai dificil să controlăm care element este în linie cu alte elemente. Folosirea poziționării relative vă permite să plasați alte elemente bazându-vă pe poziția anumitor puncte ale conținutului, cum veți vedea în ultimul exemplu al secțiunii referitoare la poziționarea relativă.

Cu puțină teorie la cingătoare, este vremea să atacăm câteva exemple de poziționare, începând cu metode absolute.

Poziționarea absolută

Voi începe cu un cod HTML care va ilustra ceea ce se mută într-o pagină. Iată un paragraf care include două intervale succesive:

```
<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 4.0//EN">
<BODY>
<P>Here is the beginning of the contents.
 <SPAN id="frontside">Here is the beginning of the frontside span.
 Here is the end of the frontside span.</SPAN>
 <SPAN id="backside">Here's the backside span...and the end of the
 backside span.</SPAN>
 Here is the end of the contents.</P>
</BODY>
```

SUGESTIE

Elementul <SPAN> este un element inline. Nu va exista nici o întrerupere suplimentară de paragraf în document, chiar dacă am adăugat ceva aici spre a face codul mai ușor de citit.

Când este vizualizat într-un browser, rezultatul este exact cum v-ați aștepta (vezi Figura 19.1). Se începe cu conținut, urmat în linie de intervalul exterior, care este urmat în linie de intervalul interior și se închide cu conținut. Fiecare element este plasat în linie relativ la predecesorul său pe pagină.

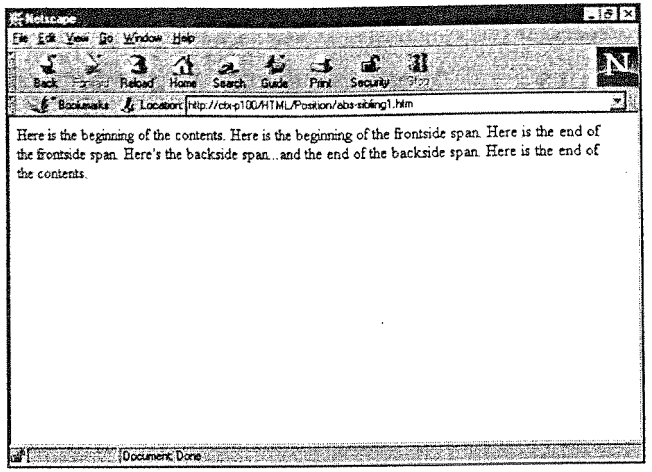


FIGURA 19.1. *Reprezentarea tradițională HTML este în linie (inline), relativ la elementul precedent.*

Pentru a modifica acest comportament, adaug o foaie de stil care deplasează intervalul din interior:

```
<STYLE type="text/css">
#frontside (position: absolute; top:150px; left:250px;
        border-style:solid; border-width:thin; border-color:black)
</STYLE>
```

Am adăugat un chenar în jurul elementului, astfel că puteți vedea spațiul pe care îl ocupă pe ecran. Reunind stilul cu textul, rezultă ceea ce vedeți în Figura 19.2.

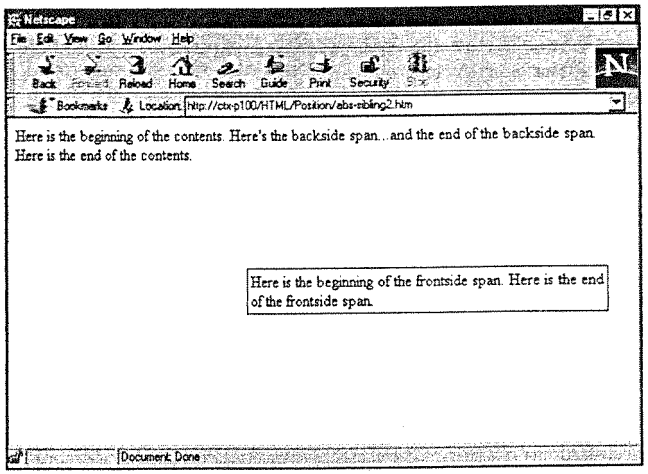


FIGURA 19.2. *Folosind poziționarea absolută, elementul exterior este plasat mai jos în dreapta.*

Acum adăugați încă o definiție de stil pentru a folosi poziționarea absolută a intervalului interior:

```
#backside {position:absolute; top:100px; left:50px;
        border-style:solid; border-width:thin; border-color:black)
```

Când noul document este afișat în browser, intervalul `backside` apare deasupra intervalului `frontside` (vezi Figura 19.3), chiar dacă acesta din urmă apare primul în cod.

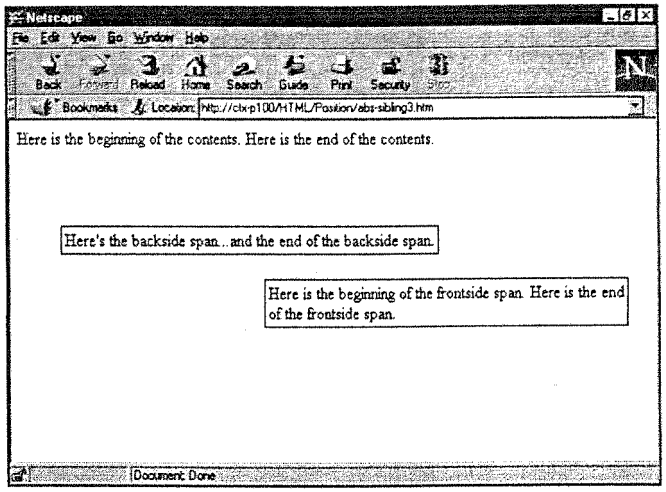


FIGURA 19.3. *Cele două intervale au fost plasate în poziții absolute în corpul documentului.*

Acum modificați codul astfel ca al doilea interval să fie în interiorul limitelor celui dintâi:

```
<BODY>
<P>Here is the beginning of the contents.
 <SPAN id="frontside">Here is the beginning of the frontside span.
 <SPAN id="backside">Here's the backside span...and the end of the
 backside span.</SPAN>
 Here is the end of the frontside span.</SPAN>
 Here is the end of the contents.</P>
</BODY>
```

Acest cod transformă `backside` într-un copil al lui `frontside`; anterior, ambele aveau același statut în corpul documentului. Rezultatul este arătat în Figura 19.4.

Nu este ceea ce așteptați. Intervalul `backside` a fost mutat acum după intervalul `frontside`. Iată explicația: poziționarea absolută este totuși relativă la *părintele* elementului. În Figura 19.3, ambele etichete `<SPAN>` sunt copii ale etichetei `<BODY>`. Prin urmare, poziționările lor absolute se bazează pe punctul de start al corpului documentului (colțul din stânga-sus al ferestrei browserului). În figura 19.4, cel de al doilea interval devine un copil al celui dintâi. Astfel, prima etichetă `<SPAN>` (`frontside`) este plasată într-o poziție absolută relativ la corp, iar a doua etichetă `<SPAN>` (`backside`) este plasată într-o poziție absolută relativ la colțul din stânga-sus al celei dintâi.

În acest moment, ați văzut principalele rezultate ale poziționării absolute:

- Se definește un punct de start nou pentru conținutul elementului.
- Poziția fizică a punctului de start este definită față de colțul din stânga-sus al *părintelui* elementului.
- Se definește un sistem de coordonate nou, care începe cu 0,0 la colțul din stânga-sus al elementului.

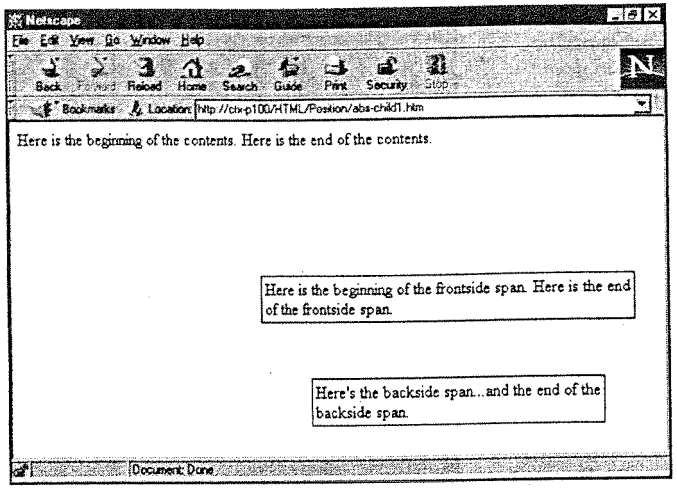


FIGURA 19.4. *Cele două intervale au poziții absolute, dar al doilea este acum copilul primului interval.*

În mod prestabilit, înălțimea elementului poziționat este exact atât de mare încât să accepte conținutul său, dar o puteți crește sau descrește cu proprietatea de stil `height`. Lățimea elementului este limitată de latura din dreapta a elementului părinte, dar și ea mai poate fi crescută cu proprietatea de stil `width`.

Acum, când posedați o înțelegere a poziționării absolute, este timpul să continuăm cu poziționarea relativă.

Poziționarea relativă

Pentru a ilustra poziționarea relativă, am de gând să încep cu un exemplu asemănător primului exemplu din secțiunea anterioară:

```
<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 4.0//EN">
<BODY>
 <SPAN id="frontside">Here is the beginning of the frontside span.
 Here is the end of the frontside span.</SPAN>
 <SPAN id="backside">Here's the backside span...and the end of the
 backside span.</SPAN>
</BODY>
```

Voi crea, apoi, o foaie de stil nouă, începând cu simpla definire a tipului de poziționare pentru al doilea interval. Rezultatul este arătat în Figura 19.5.

```
<STYLE type="text/css">
#backside {position:relative)
</STYLE>
```

Acum ajustați stilul spre a deplasa textul în jos cu 10 pixeli și la dreapta cu 30 de pixeli și reafișați rezultatele, ca în Figura 19.6:

```
#backside {position:relative; top:10; left:30)
```

Iată ce se întâmplă în Figura 19.6. Poziționarea relativă a intervalului `backside` indică browserului să mute elementul la dreapta și mai jos față de locul normal de pornire; ea *nu* plasează elementul relativ la originea elementului părinte (`<BODY>`). Aruncați o privire Figurii 19.7 spre a vedea ce se întâmplă când transformați `backside` în copil al lui `frontside`.

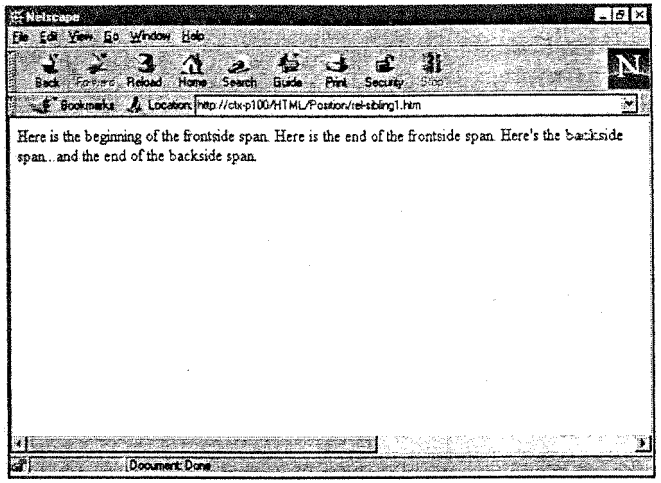


FIGURA 19.5. *Definirea poziționării celui de-al doilea interval ca relativă nu modifică documentul. Totul curge inline, cum ar trebui.*

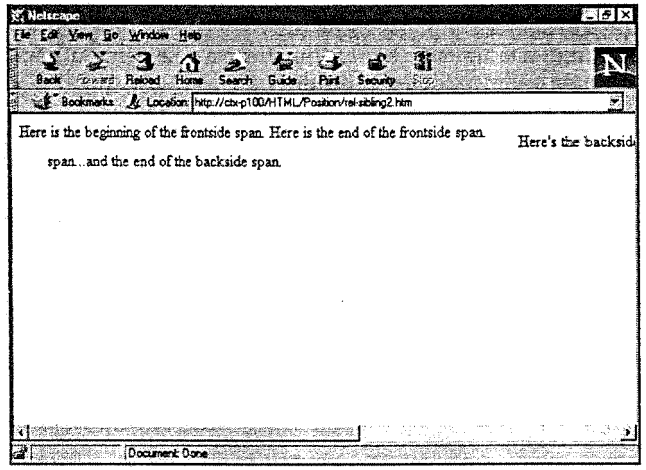


FIGURA 19.6. *Al doilea interval este deplasat la dreapta față de locul unde ar fi trebuit să pornească.*

Puteți folosi acest mod de comportare pentru efecte interesante. Aruncați o privire pe Listingul 19.1. Doriți să fiți sigur că cititorul vede numele școlii menționat în text. Pentru aceasta, deplasați paragraful la dreapta cu 25 de pixeli folosind poziționarea relativă, spre a vă oferi puțin spațiu pe partea stângă pentru a marca paragraful. Apoi, definiți un stil care să afișeze simbolurile săgeții de deplasare în spațiul liber, prin definirea unei poziții absolute la stânga paragrafului (vezi Figura 19.8).

Listingul 19.1. Folosirea poziționării pentru a plasa marcaje de paragraf în text.

```
<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 4.0//EN">

<STYLE type="text/css">
P {position:relative; left:25px}
.mark {position:absolute; left:-25px; color:white; background:gray}
</STYLE>
<BODY>
<P>We want to issue congratulations to the latest members of the
HazMat team who returned from the <SPAN class="mark">--&gt;</SPAN>
<CITE>National Fire Academy</CITE>
this summer with 80-hour certificates in Site Operating Practices of
Hazardous Materials Incidents.
</P>
</BODY>
```

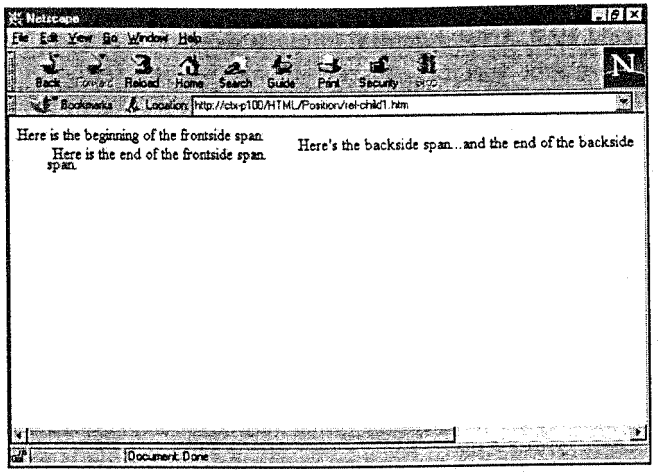



FIGURA 19.7. *Ambele părți ale intervalului părinte sunt plasate normal, pe când intervalul interior este deplasat în jos și la dreapta.*

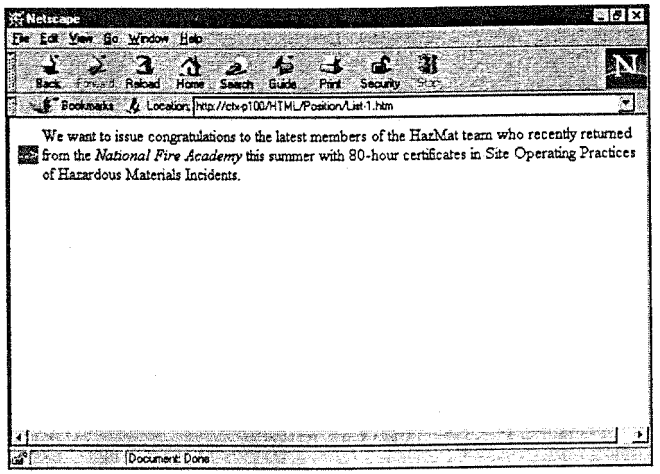


FIGURA 19.8. *Indiferent de dimensiunile ferestrei browserului, săgeata apare totdeauna la stânga paragrafului, în dreptul citatului.*

Documentul arătat în Figura 19.8 afișează o săgeată alcătuită din caractere în apropierea cuvântului evidențiat, în spațiul gol de la stânga paragrafului.

Acum ați văzut cum se comportă poziționarea relativă și cum poate fi ea folosită cu poziționarea absolută. Totuși, acest exemplu a fost numai un punct de plecare pentru

poziționare; acum este timpul să aruncăm o privire către unele dintre celelalte caracteristici ale elementelor poziționate.

Alte proprietăți ale elementelor poziționate

Acum, când v-ați jucat cu mutarea în pagină a câtorva elemente în raport unul cu altul, este timpul să vedem și alte câteva lucruri pe care elementele poziționate le pot realiza. Pentru început, încercați lucrul cu blocuri suprapuse, ca în Listingul 19.2. Rezultatele sunt arătate în Figura 19.9.

Listingul 19.2. Acest document folosește elemente suprapuse.

```
<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 4.0//EN">
<STYLE type="text/css">
#block1 {position:absolute}
#block2 {position:absolute; top:25; left:25}
</STYLE>
<BODY>
<DIV id="block1">
  <H1>This is Block 1<BR>
  This is Block 1<BR>
  This is Block 1<BR>
  This is Block 1<BR>
  This is Block 1</H1>
</DIV>
<DIV id="block2">
  <H1>This is Block 2<BR>
  This is Block 2<BR>
  This is Block 2<BR>
  This is Block 2<BR>
  This is Block 2</H1>
</DIV>
</BODY>
```

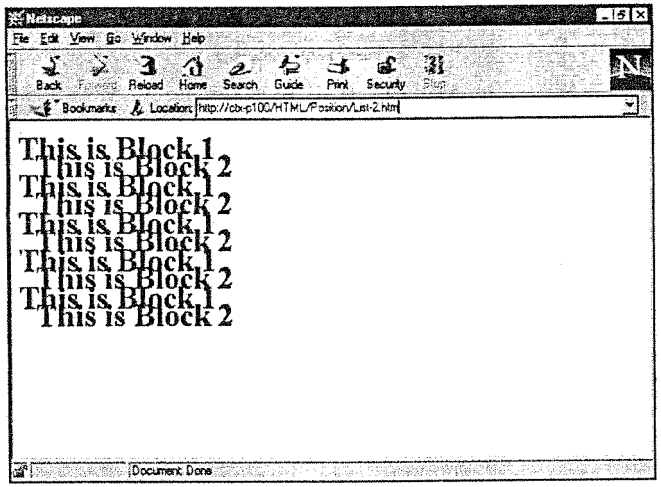


FIGURA 19.9
Blocul 2 este plasat peste blocul 1, dar încă puteți întrezări blocul 1.

Ca opțiune prestabilită, elementele poziționate sunt transparente. Pentru a trece peste acest comportament și a face ca blocul 2 să acționeze ca un perete, nu ca o fereastră, se adaugă blocului 2 o culoare de fond, cum se vede în Figura 19.10:

```
#block2 {position:absolute; top:25; left:25; background:silver}
```

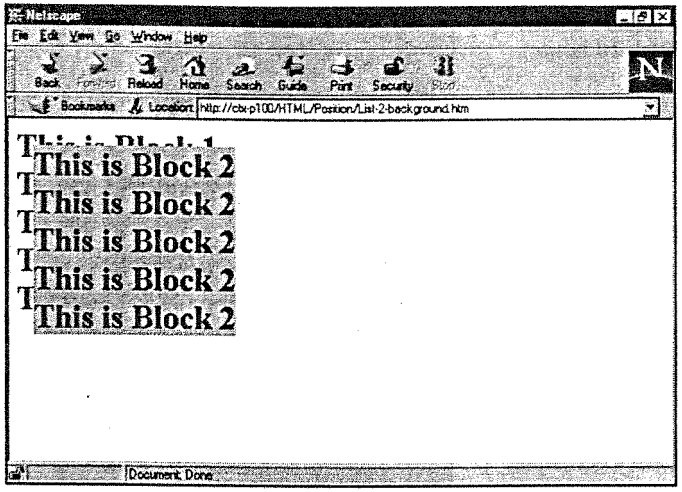


FIGURA 19.10.
Culoarea de fond a blocului 2 îl face opac, acoperind conținutul blocului 1.

De asemenea, puteți să schimbați ordinea în care sunt stivuite blocurile prin folosirea proprietății de stil `z-index`.

SUGESTIE

Termenul `z-index` reprezintă o întoarcere la zilele de școală, când trebuia să vă luptați cu geometria. Vă mai amintiți de coordonatele x, y și z ale unui spațiu tridimensional? Coordonata x este pentru orizontală, coordonata y este pentru verticală, iar coordonata z este pentru adâncime.

Pentru a ilustra acest mic artificiu, adăugați încă două blocuri paginii dumneavoastră, folosind același format ca al primelor două și adăugați niște culori de fond contrastante, astfel încât să îl puteți distinge pe fiecare separat. Fiecare bloc se suprapune peste o secțiune a celorlalte, dar ultimul bloc afișat (Blocul 4) apare totdeauna deasupra, iar Blocul 1 este totdeauna cel mai de jos (vezi Figura 19.11).

Acum adăugați proprietatea `z-index` fiecăruia dintre stiluri. Atribuirea de număr acestei proprietăți se referă la ordinea în care blocurile ar trebui să fie afișate de browser. Cu cât numărul este mai mic, cu atât mai jos este elementul în stivă. Folosind această informație, inversați ordinea de stivuire a blocurilor cu această foaie de stil:

```
#block1 {position:absolute; z-index:4; background:silver}
#block2 {position:absolute; z-index:3; background:gray;
        top:50; left:100}
#block3 {position:absolute; z-index:2; background:silver;
        color:white; top:150; left:150}
#block4 {position:absolute; z-index:1; background:black;
        color:white; top:100; left:25}
```

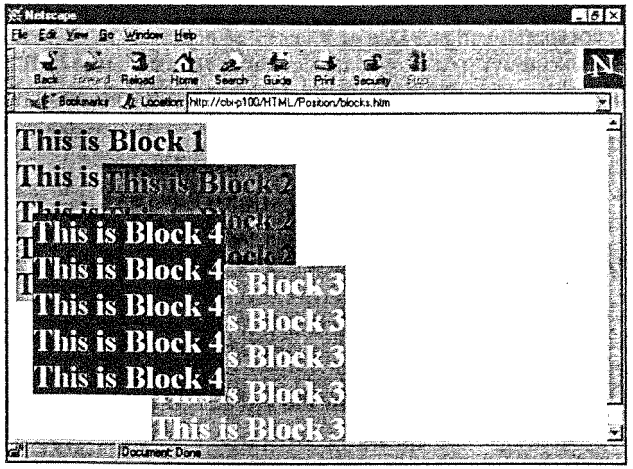


FIGURA 19.11. *Blocurile sunt plasate în pagină în ordinea în care sunt găsite în document, cu Blocul 1 la bază și Blocul 4 în vârf.*

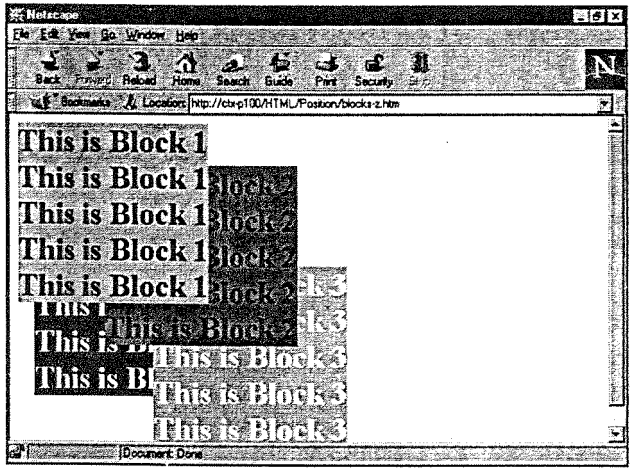


FIGURA 19.12. *Acum, Blocul 1 apare în față, iar Blocul 4 este împins în spate.*

Combinarea poziționării și a scripturilor

Această utilizare a poziționării și a scripturilor este o trăsătură care trece granițele între CSS și HTML dinamic (tratat în partea a VI-a, „HTML dinamic"). Puteți utiliza poziționarea pentru a afișa sau a ascunde elemente CSS într-o pagină. În Listingul 19.3, am folosit eticheta `<DIV>` cu o foaie de stil pentru a pune în stivă fiecare diviziune, iar apoi am controlat care din ele este în vârf cu proprietatea `z-index`. Rezultatele sunt arătate în Figura 19.13.

OBSERVAȚIE

În această secțiune, utilizez termenii *layere (straturi)*, *diviziuni* și *elemente* unul în locul altuia. Această utilizare nu implică faptul că folosesc și eticheta `<LAYER>` din Netscape, pentru că nu o fac. Totuși, regulile de poziționare pentru `<DIV>` sau `<SPAN>` se aplică și pentru `<LAYER>`; doar că Navigator 4.0 este singurul browser care o recunoaște.

Listingul 19.3. Folosirea unui script pentru a afișa diviziunea dorită și a ascunde diviziunea curentă.

```
<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 4.0//EN">
<HTML>
<HEAD>
<TITLE>Multiple Layers</TITLE>
</HEAD>

<STYLE type="text/css">
<!--
.card {position:absolute; top:75px; left:75px; width:150; background:white}
-->
</STYLE>

<SCRIPT language="javascript">
<!--
lastDiv='four';
function showLayer( thisDiv ) {
  document.all[lastDiv].style.zIndex = 1;
  document.all[thisDiv].style.zIndex = 4;
  lastDiv = thisDiv;
}
//-->
</SCRIPT>

<BODY>
<FORM>
<INPUT type="button" value="Show Layer 1" onClick="showLayer('one')">
<INPUT type="button" value="Show Layer 2" onClick="showLayer('two')">
<INPUT type="button" value="Show Layer 3" onClick="showLayer('three')">
<INPUT type="button" value="Show Layer 4" onClick="showLayer('four')">
</FORM>
<DIV id="one" class="card">
<P>This is layer number 1.
This is layer number 1.
This is layer number 1.</P>
</DIV>
<DIV id="two" class="card">
<P>This is layer number 2.
This is layer number 2.
This is layer number 2.</P>
</DIV>
<DIV id="three" class="card">
<P>This is layer number 3.
This is layer number 3.
This is layer number 3.</P>
</DIV>
<DIV id="four" class="card">
<P>This is layer number 4.
This is layer number 4.
This is layer number 4.</P>
</DIV>
</BODY>
</HTML>.
```

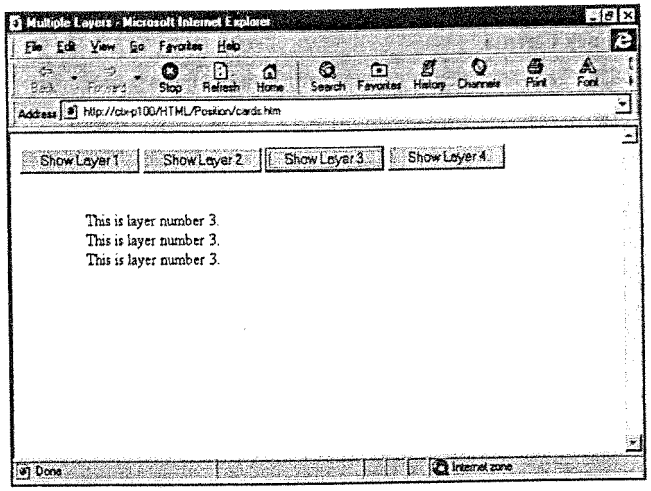


FIGURA 19.13. *Folosind CSS, scripturi și etichete <DIV>, ați creat un efect simplu de HTML dinamic cu poziționare CSS.*

Când se execută clic pe un buton (precum Show Layer 2 – Arată Layer 2), este apelată o funcție JavaScript (`showLayer`) pentru a aloca un `z-index` inferior diviziunii afișate curent (1, dacă pagina tocmai a fost încărcată) și un `z-index` superior noii diviziuni (în acest caz, 2). Același proces continuă cu noua diviziune și diviziunea care este selectată prin apăsarea unui buton.

Rezumat

Poziționarea CSS oferă autorilor un element de control suplimentar asupra paginilor lor Web. Nu mai sunt îngrădiți de constrângerea asupra locului unde apare o bucată de text în ordinea documentului. Folosind poziționarea, puteți face ca elementul să apară aproape oriunde doriți.

Există două tipuri de poziționare: relativă și absolută. Poziționările relative depind de locul în care elementul ar fi fost redat în fluxul natural al documentului. Poziționările absolute se bazează pe colțul din stânga-sus al elementului părinte.

Toate sistemele de coordonate în poziționarea CSS se bazează pe colțul din stânga-sus al elementului părinte, începând cu corpul documentului. Aceasta deoarece colțul din stânga-sus este singurul punct fix pe care vă puteți baza într-un document. Colțul din dreapta-jos depinde totdeauna de browserul, platforma și configurațiile particulare ale utilizatorului.

Folosirea poziționării într-un document vă permite, de asemenea, să suprapuneți elemente pe o pagină. O dată suprapuse, ordinea de stivuire poate fi schimbată cu proprietatea `z-index`. Combinând poziționarea și stivuirea cu scripturi, este posibil să producem efecte elementare de HTML dinamic.

Capitolul 20
Crearea cadrelor

de Rick Darnell

În acest capitol

- **Marea dispută asupra cadrelor 274**
- **Crearea unui set de cadre 274**
- **Comunicarea între cadre 278**
- **Introducerea unui conținut alternativ în cadre 281**
- **Dincolo de ferestrele împărțite: cadrul inline 282**
- **Două machetări obișnuite de cadre 283**

Cadrele reprezintă implementarea unui concept introdus mai întâi de către Netscape în 1995 prin lansarea produsului lor Navigator 2.0. După multe dezbateri și frământări de mâini, Consorțiul World Wide Web (W3C) a decis să includă cadrele în standardul HTML 4.0.

Cadrele, în cazul în care nu le-ați experimentat, vă permit să împărțiți fereastra browserului în segmente mai mici și să afișați câte o pagină Web diferită în fiecare dintre ele (vezi Figura 20.1). Cadrele pot interacționa între ele, acțiunile de pe o pagină dintr-un cadru afectând conținutul și comportamentul unui alt cadru.

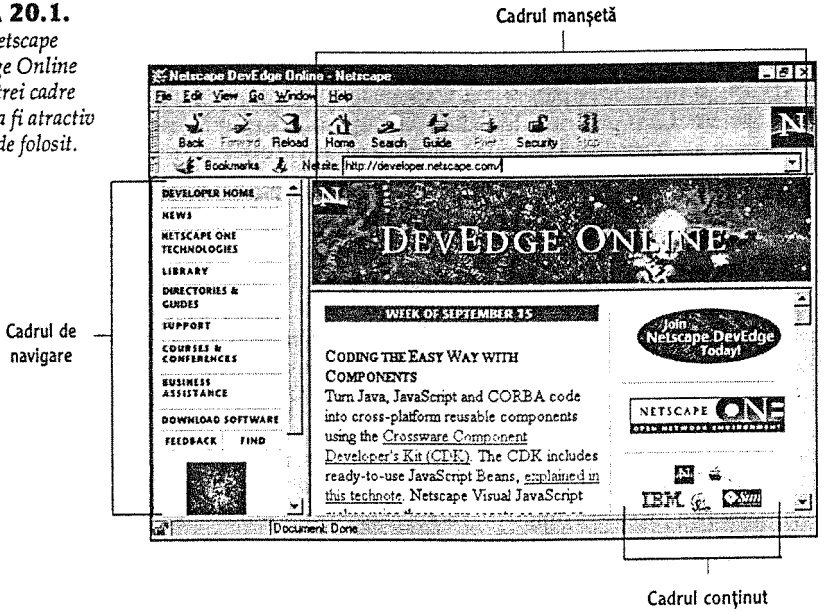


FIGURA 20.1. *Situl Netscape DevEdge Online posedă trei cadre pentru a fi atractiv și ușor de folosit.*

Acest capitol vă introduce în mecanismul creării de cadre în cele două ipostaze fundamentale: drept parte intrinsecă a ferestrei browserului și drept conținut inline.

Marea dispută asupra cadrelor

W3C a dus o sumedenie de lupte pentru introducerea cadrelor ca parte a standardului HTML 4.0. Majoritatea autorilor și a beneficiarilor privesc cadrele sau cu dragoste, sau cu ură. Mulți autori le iubesc pentru flexibilitatea deosebită pe care o oferă în distribuirea cantităților mari de conținut către utilizator, dar multor utilizatori nu le plac datorită timpului și resurselor suplimentare pe care le reclamă, fără a menționa și limitarea la browserele care acceptă cadre.

Cel mai important argument împotriva cadrelor este că ele controlează aspectul browserului utilizatorului, ceea ce contravine scopului HTML de standard independent de dispozitiv și de prezentare.

Una dintre soluțiile alternative la cadre, care a fost acceptată de HTML 3.0 și a fost preluată în HTML 3.2, este eticheta <BANNER>. Această etichetă era folosită spre a indica un element asemănător cadrului în partea de sus sau de jos a ecranului, care putea fi folosit pentru reclamă sau legături, fără a monopoliza o seamă de resurse ale browserului.

Argumentul pentru <BANNER> și împotriva etichetei <FRAME> nu este implementarea propriu-zisă, ci modul în care a fost realizată ea în multe situri Web. Detractorii cadrelor pretind că majoritatea siturilor care înglobează cadre o fac într-o manieră foarte neprietenoasă față de cititor, prin forțarea întâietății machetei autorului asupra cititorului.

Oponenții merg și mai departe, pretinzând că mulți autori din Web refuză să înglobeze date <NOFRAME> semnificative în documentul lor, altele decât comentariul „Aveți nevoie de un produs Netscape / Microsoft", ceea ce nu ajută oamenii pe unele platforme sau cu probleme de accesibilitate.

Având în vedere aceste argumente și multe altele, sunt cadrele o adăugare inerent nefastă la HTML? După umila mea părere, nu. Capcanele lor ar putea fi mult mai înșelătoare pentru utilizatorul final, dar pentru autorii din Web responsabili există o mulțime de moduri de a se asigura cum se degradează o pagină cu cadre pentru browserele necompatibile.

Metodele de a realiza o degradare elegantă sunt incluse în descrierea fiecărui element de cadru, iar folosirea sa în secțiunile care urmează.

Crearea unui set de cadre

Dacă începeți să creați un set de cadre pentru situl dumneavoastră Web, este de ajutor să vă acordați câteva minute pentru a vă întreba de ce le folosiți, în primul rând. Cadrele sunt instrumente folositoare, dar, așa cum nu are rost să utilizați un baros spre a construi o casă de păpuși, cadrele nu sunt totdeauna indicate pentru oricare sit Web.

Amintiți-vă, cadrele reprezintă, încă, o adăugire relativ nouă pentru Web. Unele browsere obișnuite, precum Mosaic, Lynx și versiuni mai vechi de Navigator și Internet Explorer, nu acceptă cadrele. Ca rezultat, dacă folosiți exclusiv cadre, utilizatorii de browsere incompatibile ajung la a blestema timpul în care navighează prin situl dumneavoastră.

Dar să presupunem că ați reflectat asupra chestiunii și acum ați decis să realizați un mijloc de navigare prin Web cu cadre. Este subtil și de efect; tot ceea ce aveți de făcut este să adăugați un set cu câteva cadre sitului dumneavoastră și veți fi gata de drum.

<FRAMESET>

Adăugarea unui set de cadre reprezintă o simplă chestiune de includere a etichetelor containerului <FRAMESET> după antetul documentului în locul corpului:

```
<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 4.0//EN">
<HTML>
<HEAD>
  <!--Header material-->
</HEAD>

<FRAMESET>
  <!--Frame definitions-->
</FRAMESET>

</HTML>
```

În cazul browserelor care acceptă cadre, conținutul setului de cadre este analizat și interpretat în consecință. În cazul browserelor fără cadre, rezultatul este o pagină goală, pe care o vom adresa într-o clipă cu <NOFRAMES>. Pentru moment, sintaxa pentru elementul setului de cadre este aceasta:

```
<FRAMESET [rows=rowWidths][cols=colWidths][loadEvents]>
  ...Frame or Frameset definitions...
</FRAMESET>
```

Aici, *rowWidths* și *colWidths* reprezintă o listă de dimensiuni pentru respectivele linii sau coloane de cadru, delimitate de virgulă. Dacă nu sunt incluse, fiecare valoare se prestabilește la un rând și o coloană, indiferent de numărul de cadre care sunt incluse în definiție. Eticheta <FRAMESET> acceptă și evenimentele onLoad și onUnload pentru a detecta când au fost încărcate toate cadrele și, respectiv, când utilizatorul părăsește acest set de cadre.

SUGESTIE

Este important de semnalat aici că o singură etichetă <FRAMESET> poate include una sau mai multe etichete adiționale <FRAMESET> între hotarele sale. Aceste etichete adiționale vă permit să creați machete cadru mai complexe, cum veți vedea în secțiunea următoare, despre eticheta <FRAME>.

De exemplu, un set de cadre cu trei linii care ocupă 20, 20 și 60% din spațiul disponibil pentru fereastră ar putea fi definit după cum urmează:

```
<FRAMESET rows="20%,20%,60%">
  ...Frame definitions...
</FRAMESET>
```

Sau ați putea defini spațiile dintre linii folosind notația proporțională:

```
<FRAMESET rows="1*,1*,3*">
  ...Frame definitions...
</FRAMESET>
```

Acest cod indică un set de cadre cu trei linii care ocupă, respectiv, 1/5, 1/5 și 3/5 din spațiul disponibil. Pentru a defini spațiul în pixeli, omiteți toate sufixele pentru măsuri.

Întrucât, anterior în această secțiune, am menționat posibilitatea etichetelor <FRAMESET> imbricate, să aruncăm o privire asupra modului în care se înfăptuiește aceasta. Ca un exemplu, în următoarea secțiune sunteți îndrumat spre a crea o pagină care include un mic cadru orizontal deasupra unui cadru de coloană îngustă și a unui cadru de coloană largă. Pentru a defini această structură, începeți cu liniile:

```
<FRAMESET rows="20%,80%">
</FRAMESET>
```

Apoi, adăugați un cadru pentru prima linie și două coloane în linia a doua. Nu duceți grija atributului etichetei <FRAME>; este tratat în secțiunea următoare.

```
<FRAMESET rows="20%,80%">
  <FRAME scrolling="yes">
  <FRAMESET cols="25%,75%">
    <FRAME scrolling="yes">
    <FRAME scrolling="yes">
  </FRAMESET>
</FRAMESET>
```

Observați că nu există încă un fișier sursă pentru conținutul cadrelor. Browserul dumneavoastră ar fi trebuit să creeze doar un set gol de cadre. Veți vedea cum să precizați sursa într-o clipă. Rezultatul final este arătat în Figura 20.2.

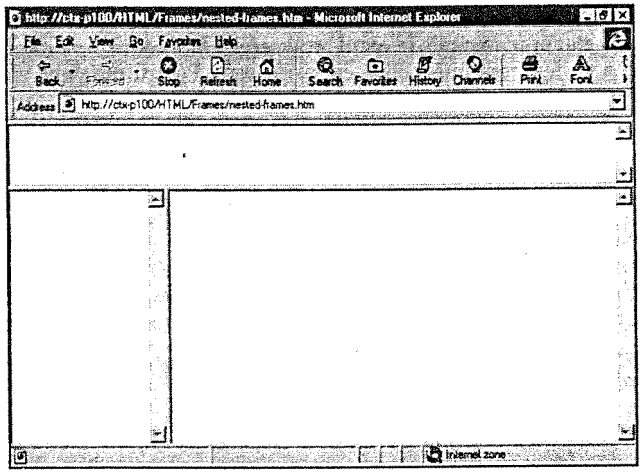


FIGURA 20.2. *Rezultatul seturilor de cadre imbricate este un rând lung peste o coloană îngustă și o coloană mai largă.*

Revenind, acest exemplu împarte întâi browserul în două rânduri; rândul de sus ocupă 20% din spațiu și cel de jos are 80% din spațiu. Apoi, rândul al doilea este împărțit în două coloane, coloana din stânga ocupând 25% din spațiu și coloana din dreapta deținând 75% din spațiu. Aceasta înseamnă că în dreapta-jos se află cadrul cel mai mare, cu 60% din aria totală a ferestrei (75% din spațiul orizontal multiplicat cu 80% din spațiul vertical).

OBSERVAȚIE

De fapt, spațiul disponibil pentru afișare este puțin mai mic, deoarece există rezervări pentru bare de defilare.

Dacă includeți o etichetă <FRAMESET> în pagina dumneavoastră, puteți continua să imbricați etichete <FRAMESET> suplimentare până când vă albăstriți la față și nu mai rămâne loc pe ecran. În cele din urmă, aveți nevoie să adăugați o etichetă <FRAME> pentru a comunica browserului care ar trebui să fie conținutul fiecărui spațiu.

<FRAME>

Raportul dintre <FRAME> și <FRAMESET> se aseamănă mult cu raportul dintre un articol de listă și containerul listă (vezi Capitolul 8, „Folosirea listelor pentru organizarea informațiilor"), exceptând faptul că aceste cadre definesc și spațiul vizibil de pe browser.

Sintaxa pentru eticheta <FRAME> este aceasta:

```
<FRAME [name=][src=][behavior][appearance]>
```

Aici, `name` este un nume specificat de utilizator pentru cadrul particular, iar `src` este adresa URL inițială de afișat în interiorul cadrului. Pentru a modifica ulterior un cadru spre a căpăta un comportament personalizat, citiți în următoarele două secțiuni despre atributele care îl afectează.

Comportamentul de bază al cadrului

Două atribute controlează comportarea fundamentală a unui cadru: `scrolling` și `noresize`. Aceste atribute stabilesc dacă respectivul cadru oferă capacitatea de defilare pentru beneficiar și dacă beneficiarul poate ajusta proporțiile cadrului.

- **`scrolling`:** Controlează folosirea barelor de defilare pe cadru. Când conținutul cadrului depășește lățimea sa, cele mai multe browsere adaugă bare de defilare, astfel încât utilizatorul poate vedea întregul conținut. Această trăsătură este folositoare când pagina conține imagini sau o formatare care o face să se extindă dincolo de marginile ei stabilite (vezi secțiunea următoare, „Aspectul cadrului", pentru informații despre stabilirea dimensiunii și marginilor cadrului).

Cele trei posibilități pentru `scrolling` sunt `yes`, `no` și `auto`. Valoarea prestabilită este `auto`. Pentru `scrolling=yes`, totdeauna browserul furnizează bare de defilare sau alte instrumente adecvate, fie că ele sunt într-adevăr necesare sau nu. Dacă nu sunt necesare pentru conținut, atunci browserul ar trebui să le dezactiveze. O valoare `no` semnifică faptul că dispozitivele de defilare nu sunt niciodată folosite, adică un conținut peste dimensiuni este retezat, afișându-se atât cât încape, începând cu colțul din stânga-sus al fișierului.

- **`noresize`:** Folosirea acestui atribut oprește utilizatorul de la redimensionarea cadrelor individuale. Această caracteristică este efectivă atunci când conținutul depinde de o anumită dimensiune a cadrului, ca în cazul unui cadru cu imagini. În caz contrar, browserul folosește în mod prestabilit cadre redimensionabile.

Aceste două atribute acoperă fundamentele unui cadru. Totuși, pentru un control extins asupra acestor elemente, ați putea dori să luați în considerare folosirea atributelor de aspect, definite în secțiunea următoare.

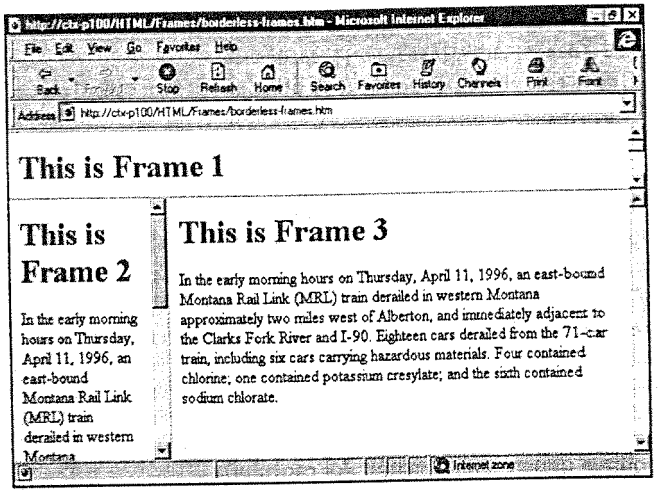
Aspectul cadrului

Trei atribute controlează aspectul cadrului pentru utilizator: `frameborder`, `marginwidth` și `marginheight`.

- **`frameborder`:** Controlează dacă respectivul cadru este înconjurat de un chenar pentru a-l separa vizual de restul conținutului. Pentru a include un chenar, utilizați `frameborder=1`; altfel, utilizați `frameborder=0`. Cele mai multe browsere folosesc chenarul în mod prestabilit. Dacă adăugați atributul `frameborder` exemplului de set de cadre imbricate pentru a ascunde chenarele, veți obține un exemplu care are bare de defilare, dar nici o linie despărțitoare (vezi Figura 20.3).
- **`marginwidth`:** Acest atribut stabilește dimensiunea spațiului, specificată în pixeli, dintre latura stângă și cea dreaptă și conținutul cadrului. Valoarea stabilită cu acest atribut trebuie să fie un întreg, ca `1` sau mai mare. Valoarea prestabilită este dată de browserul respectiv.
- **`marginheight`:** Asemănător cu `marginwidth`, acest atribut stabilește spațiul dintre latura de sus și cea de jos și conținutul cadrului. El acceptă aceleași valori ca și `marginwidth`.

Acum, când ați stabilit comportamentul de bază și aspectul cadrelor, este timpul să stabiliți un mod în care cadrele să comunice și să interacționeze unul cu celălalt.

FIGURA 20.3. *Setul de cadre imbricate include bare de defilare, dar nici un chenar.*



Comunicarea între cadre

Metoda principală de a permite unui cadru să comunice cu altul este prin folosirea atributului `target`. Acest atribut este disponibil pentru etichete care afectează hiperlegături (precum `<A>` și `<LINK>`), hărți imagine (`<AREA>`), formulare (`<FORM>`) și elementul `HEAD` (`<BASE>`).

Pentru a înțelege atributele `target` și cum să lucrați între cadre, veți folosi un set de cadre simplu, numit `frames.htm` (vezi Listingul 20.1).

Listingul 20.1. Sursa HTML pentru `frames.htm`.

```
<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 4.0//EN">
<FRAMESET rows="50%,50%">
  <FRAME name="toprow" src="page1.htm">
  <FRAME name="botrow" src="page2.htm">
</FRAMESET>
```

Nu, nu este o pagină cu cadre degradabilă, dar nu acesta este scopul imediat. În continuare, veți avea nevoie de un cod sursă pentru primele două pagini, arătat în listingurile 20.2 și 20.3.

Listingul 20.2. Sursa HTML pentru cadrul `toprow` inițial: `page1.htm`.

```
<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 4.0//EN">
<BODY>
<H1>This is page 1.</H1>
<P><A href="page2.htm">This is a link to page 2.</A></P>
<P><A href="page3.htm">This is a link to page 3.</A></P>
<P><A href="page4.htm">This is a link to page 4.</A></P>
</BODY>
```

Listingul 20.3. Sursa HTML pentru cadrul `botrow` inițial: `page2.htm`.

```
<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 4.0//EN">
<BODY>
```



```
<H2>This is the end of page 2.</H2>
<H2>This is the end of page 2.</H2>
<H2>This is the end of page 2.</H2>
</BODY>
```

Codurile sursă HTML pentru paginile 3 și 4 sunt aceleași ca `page2.htm`, exceptând numele de fișiere și numerele de pagină. Când `frames.htm` este încărcat inițial într-un browser, rezultatul este ceea ce vedeți în Figura 20.4. Observați ce se întâmplă când executați clic asupra oricăreia dintre hiperlegăturile din cadrul de sus. Cadrul de sus se schimbă, dar nimic nu se întâmplă cu cadrul de jos (vezi Figura 20.5).

FIGURA 20.4. *Rezultatul încărcării fișierului `frames.htm` în Internet Explorer.*

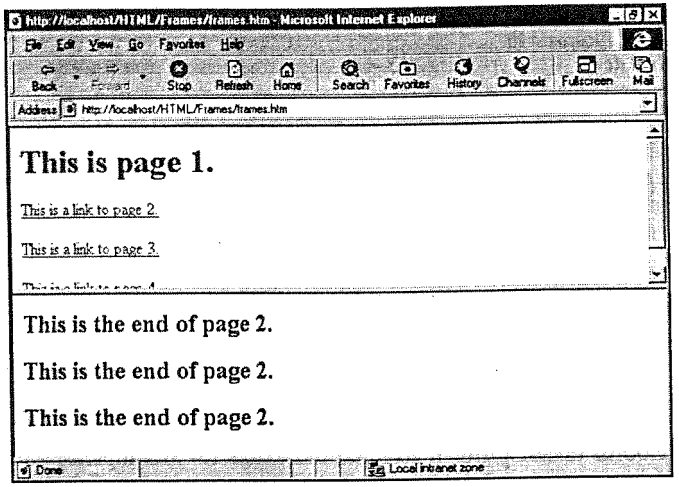
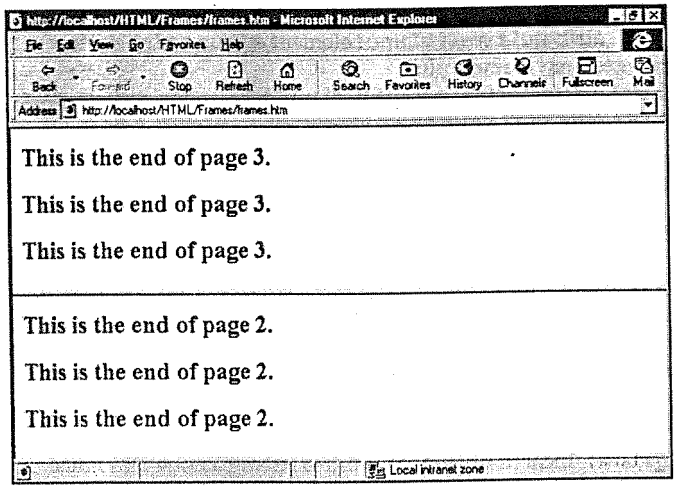


FIGURA 20.5. *Ținta unei hiperlegături este încărcată în cadrul de sus, unde a fost selectată.*



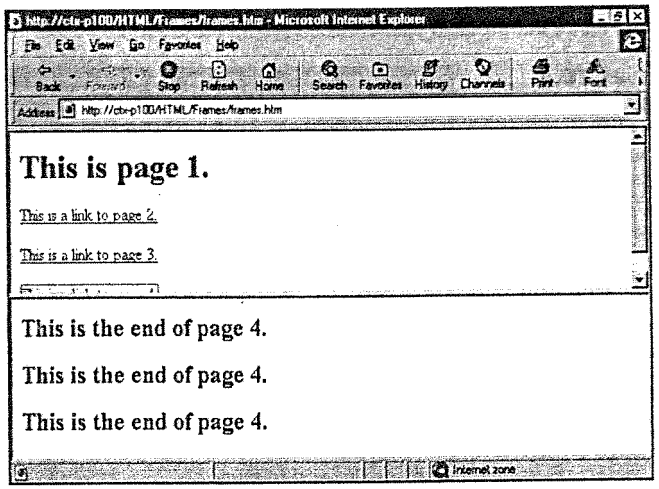
Felul în care doriți ca această pagină să lucreze este ca ținta unei hiperlegături din cadrul de sus să se încarce în cadrul de jos; iar acesta este locul în care atributul `target` intră în

acțiune. Modificați `page1.htm` prin adăugarea etichetei `<BASE>` cu o anume țintă la începutului documentului:

```
<HEAD>
<BASE target="botrow">
</HEAD>
```

Acum, când reîncărcați setul de cadre și executați clic pe o hiperlegătură din fereastra de sus, conținutul apare în fereastra de jos (vezi Figura 20.6).

FIGURA 20.6. *Ținta unei hiperlegături din cadrul de sus este încărcată în cadrul de jos.*



Atributul `target` putea fi adăugat individual fiecărei hiperlegături în prima pagină, dar adăugarea o singură dată a elementului `BASE` a fost mai mult decât suficientă. Câteva reguli suplimentare guvernează comportarea țintelor și cadrelor:

- Dacă atributul `target` al unui element este un cadru existent, atunci acel cadru primește ținta elementului.
- Dacă un element nu posedă atributul `target`, dar eticheta `<BASE>` o face, atunci eticheta `<BASE>` determină destinația conținutului.
- Dacă atât elementul, cât și eticheta `<BASE>` includ `target` și fiecare are valori diferite, atunci se folosește valoarea elementului, iar `<BASE>` este ignorată.
- Dacă nici elementul, nici eticheta `<BASE>` a documentului nu menționează o țintă, atunci cadrul curent, care conține elementul activat, primește conținutul.
- Când cadrul denumit specificat în `target` nu există, atunci se creează o nouă fereastră și i se atribuie acel nume necunoscut, iar conținutul este încărcat în noua fereastră.

De asemenea, există unele nume `target` speciale pe care le puteți folosi și care sunt identificate prin caracterul liniuță de subliniere la început (_). Ele sunt nume rezervate, care nu pot fi folosite cu atributul `name`. Numele rezervate identifică relațiile dintre seturile de cadre, cadrele lor și seturile de cadre imbricate. Iată numele și efectele lor când sunt utilizate ca atribute `target`:

- **`_blank`:** Încarcă documentul într-o fereastră nouă fără nume.
- **`_self`:** Încarcă documentul în cadrul elementului care a folosit acest nume.
- **`_parent`:** Încarcă documentul în setul cadru care conține cadrul curent.
- **`_top`:** Încarcă documentul în fereastra originală, renunțând la toate celelalte cadre.

Veți vedea variatele utilizări pentru `target` atunci când vom crea ierarhii imbricate de cadre în ultima secțiune a acestui capitol, „Două machetări obișnuite de cadre".

Introducerea unui conținut alternativ în cadre

Acum, când ați definit seturile de cadre și cadrele, este vremea să luați în considerare mulțimea celor fără browsere care acceptă cadre.

OBSERVAȚIE

W3C îndeamnă ferm autorii să prevadă un conținut alternativ pentru browserele care nu acceptă cadre sau care sunt configurate pentru a le ignora. Sunt întru totul de acord. Utilizarea celor mai recente caracteristici este nemaipomenită, dar nu uitați: persoanele care folosesc ce este mai recent și ce este mai grandios reprezintă încă unul dintre cele mai mici segmente ale comunității utilizatorilor de browsere.

Există două metode de adaptare la browserele fără cadre. Prima este folosirea etichetei `<FRAMESET>` însăși. Amintiți-vă că setul de cadre înlocuiește corpul documentului. Utilizatorii fără browsere compatibile cu cadrele nu vor vedea nimic atunci când încarcă pagina dumneavoastră. Totuși, puteți adăuga un conținut pentru browserele necompatibile cu ajutorul etichetei `<NOFRAMES>`, plasată după ultima etichetă `<FRAME>`. Puteți insera orice cod HTML valid în interiorul unui set de `<NOFRAMES>`, cum ar fi o pagină de bază alternativă sau o legătură către un alt punct de start în situl dumneavoastră Web.

```
<FRAMESET cols="25%,75%">
  <FRAME name="toc" src="home-toc.htm">
  <FRAME name="content" src="home.htm">
  <NOFRAMES>
    Thanks for stopping at our Web site. To view this site without
    frames, check out our <A href="home.htm">no-frame Home Page</A>.
  </NOFRAMES>
</FRAMESET>
```

Dacă punctul de start pentru situl dumneavoastră fără cadre este și sursă pentru unul din cadrele dumneavoastră, atunci puteți folosi eticheta `<NOFRAMES>` pentru a include bucăți de conținut din alte pagini cu cadre.

Să considerăm fragmentul de set de cadre anterior. Dacă ați lăsa tabla de materii (TOC = table of contents) și pagina de bază separate, atunci utilizatorul fără un browser cu cadre nu ar beneficia de folosirea paginii cu TOC. Utilizând `<NOFRAMES>`, puteți include TOC în pagina de bază fără a o afișa când se folosesc cadrele. Următorul fragment este din pagina de bază fictivă:

```
<!-- the home page -->
<BODY>
<NOFRAMES>
...table of contents here...
...table of contents here...
...table of contents here...
</NOFRAMES>
<H1>The Home Page</H1>
...the main home page content goes here...
</BODY>
```

Când pagina este vizualizată într-un browser cu cadre, totul se potrivește planului, după cum ați văzut în exemplele precedente de cadre, iar tabla de materii din pagină este

ascunsă. Când este văzută într-un browser fără cadre, tabla de materii înglobată în pagină este dezvăluită.

ATENȚIE!

Deși folosirea precedentă a etichetei <NOFRAMES> este conformă standardului HTML 4.0, ea nu este acceptată în mod obișnuit de către vasta majoritate a browserelor, inclusiv Navigator și Internet Explorer.

Folosind aceste două caracteristici ale paginilor cu cadre, puteți crea un sit Web care este funcțional și utilizabil de către *toți* beneficiarii dumneavoastră, nu doar de aceia care folosesc ultima versiune de Navigator sau Internet Explorer.

Mai există încă o opțiune pentru cadre de tratat în acest capitol: cadrele pe port de conținut în interiorul unei pagini Web.

Dincolo de ferestrele împărțite: cadrul inline

Cadrele inline sunt un pas înainte relativ nou în lumea cadrelor. Un cadru inline este, în esență, tratat în același fel ca o imagine sau un obiect inline; el ocupă un bloc de spațiu pe pagină pentru inserarea de conținut specializat (în acest caz, o altă pagină Web).

ATENȚIE!

În mod curent, Internet Explorer (versiunile 3 și 4) este singurul browser care acceptă cadrele inline. Toate celelalte browsere ignoră acest element.

Cadrele inline sunt folositoare pentru a plasa lucruri ca indexuri sau texte suplimentare, care contribuie la documentul curent, dar nu sunt cruciale. De asemenea, folosirea cadrelor inline pentru informații suplimentare face mai ușor de introdus un conținut alternativ pentru browserele incompatibile.

Sintaxa pentru un cadru inline (<IFRAME>) este aceasta:

```
<IFRAME [name=frameName][src=URL][frameborder=1or0][marginwidth=numPixels]
  [marginheight=numPixels][scrolling=yesOrNo][align=verticalOrHorizontal]
  [height=frameHeight][width=frameWidth]>
</IFRAME>
```

Atributul `name` este folosit pentru a da cadrului un nume care poate fi folosit de către alte hiperlegături în documentul curent, iar `src` este adresa URL a paginii Web sau a altui conținut care va umple cadrul. Restul atributelor pentru cadrul inline au fost tratate sub eticheta <FRAME> mai înainte în acest capitol, excepând `align`, `height` și `width`.

Atributul `align` este un vechi prieten din zilele lucrului cu <IMG>. Valorile posibile sunt `bottom`, `middle`, `top`, `left` sau `right` (jos, mijloc, sus, stânga sau dreapta). Valoarea prestabilită este `bottom`, care aliniază partea de jos a cadrului cu linia de bază curentă a textului unde este localizată.

În același mod, `middle` aliniază punctul de mijloc al cadrului pe verticală cu linia de bază a textului, iar `top` aliniază partea de sus a cadrului cu partea de sus a liniei de text. Ultimele două, `left` și `right`, forțează cadrele la marginile respective și fac ca textul să curgă în jurul cadrelor.

OBSERVAȚIE

Deși `align` nu mai este recomandat în aproape toate celelalte elemente din HTML 4.0, el pare că rămâne în utilizarea standard pentru <IFRAME>. Versiunea preliminară HTML 4.0 Document Type Definition (Definiția modelului de document HTML 4.0) nu include suport pentru `class` sau `style`. Totuși, este credința mea că browsere ca Navigator și Internet Explorer vor mai accepta aceste opțiuni, iar W3C va corecta această aparentă scăpare din vedere la următoarea revizie HTML.

Dacă nu se specifică nici o valoare pentru `height` și `width`, atunci browserul face tot posibilul să aloce suficient spațiu pentru conținut. Altfel spus, puteți menționa o valoare fie pentru una, fie pentru ambele folosind tehnicile celor trei măsuri standard: pixeli (`height="100"`), procente (`width="20%"`) sau un raport (`width="2*"`).

Ca un exemplu de cadru inline, codul HTML care urmează vă oferă un document sursă care include câteva informații adiționale într-un cadru inline (vezi Figura 20.7). Dacă nu sunt acceptate cadrele inline, atunci browserul ignoră eticheta <IFRAME> și afișează hiperlegătura pentru conținut (vezi Figura 20.8).

```
...content...
<IFRAME src="techRequirements.htm" width="50%" height="200"
        scrolling="auto" align="right" frameborder="1">
  <!--User agent does not support frames-->
  Interested in being a HazMat Technician? Then check out the
  <A href="techRequirements.htm" title="Technician Requirements">
HazMat Team Requirements</A>.
</IFRAME>
...more content...
```

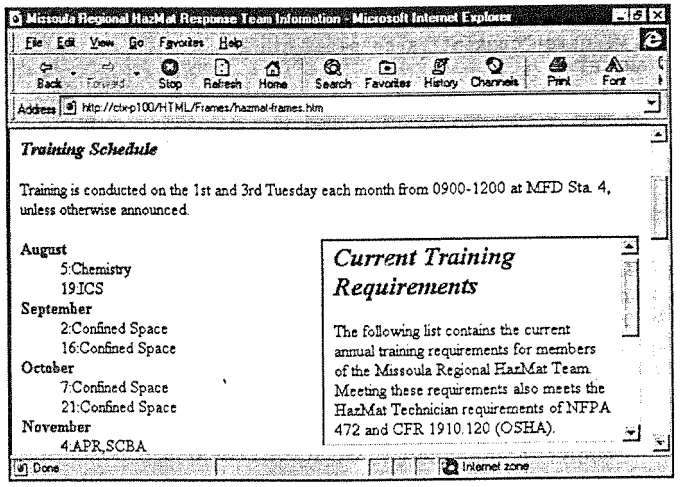
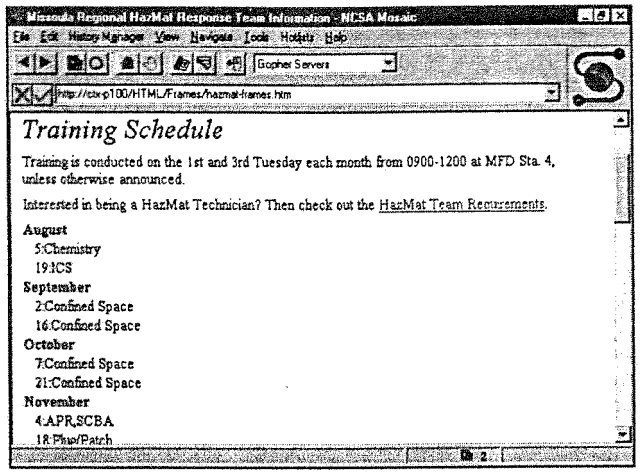


FIGURA 20.7. *Într-un browser compatibil cu cadrele, precum Internet Explorer, cadrul inline este prezentat în mijlocul celuilalt conținut al paginii.*

Două machetări obișnuite de cadre

Când porniți lucrul cu cadre, poate fi dificil să estimați cum să le implementați efectiv. Următoarele secțiuni tratează câteva dintre cele mai folositoare și obișnuite machete de cadru.

FIGURA 20.8. *Într-un browser fără cadre, precum Mosaic, pagina se degradează și oferă o legătură către pagina care ar fi apărut în cadru.*

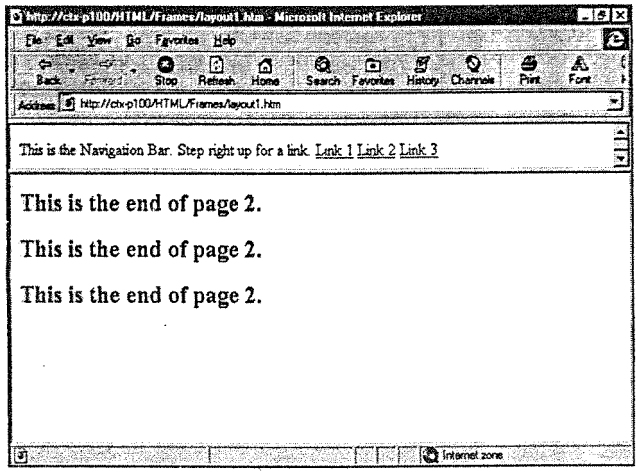


Bară de navigație cu conținut

Bara de navigație cu conținutul într-un set de cadre constă dintr-o linie sau coloană îngustă cuprinzând legături de navigare în sit și un cadru mai larg cu conținut principal (vezi Figura 20.9):

```
<!--navbar at top, content on bottom-->
<FRAMESET rows="15%,85%">
  <FRAME name="navbar" src="navbar.htm">
  <FRAME name="content" src="page2.htm">
</FRAMESET>

<BODY>
Welcome to the XYZ Web site. To continue, please go to our no-frames
<A href="index-noframe.htm" title="Frameless Home Page">Home Page</A>.
</BODY>
```

FIGURA 20.9. *Una dintre machetările de bază pentru un set de cadre: index în partea de sus și cuprins în partea de jos.*



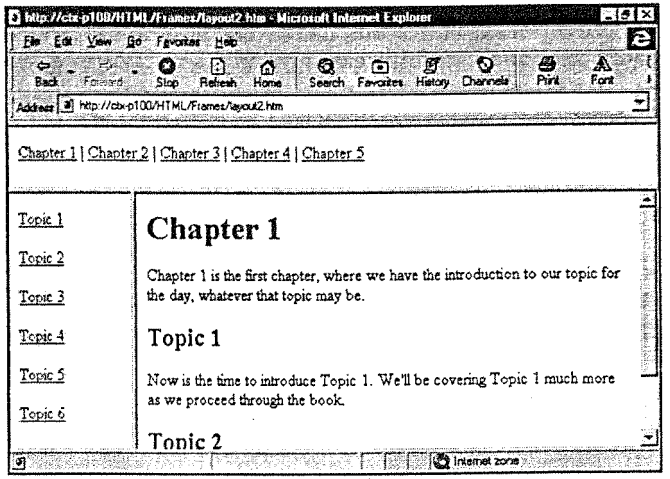
Cadrul de navigare poate fi plasat de-a lungul oricărei laturi a conținutului, în funcție de destinație. De exemplu, plasarea cadrului `navbar` în partea de jos a setului de cadre este folositoare pentru a include un set de note de subsol pentru documentul curent. Fișierul `navbar.htm` trebuie să aibă `<BASE target="content">` în fruntea documentului pentru a păstra cadrul de navigare în timp ce afișează cuprinsul în cadrul pentru conținut.

Pentru a arăta acest cadru ca pe niște coloane verticale, înlocuiți atributul `rows` cu atributul `cols`. Alte utilizări pentru această machetare includ plasarea unei reclame în coloana îngustă sau includerea unui index pentru cadrul principal al documentului.

Tabla de materii, trecerea în revistă a capitolelor și cuprinsul

Clădind scheletul precedentului exemplu, ați recurs la a adăuga încă un cadru ca o coloană dedesubtul manșetei din partea de sus. Această structură vă oferă o ierarhie imbricată, în care cadrul din partea de sus controlează conținutul coloanei din stânga, iar coloana din stânga controlează coloana de pe dreapta (vezi Figura 20.10).

FIGURA 20.10. *Aceste cadre reprezintă o ierarhie imbricată; partea de sus controlează partea din stânga și partea din stânga pe cea din dreapta.*



Această machetă de cadru este creată în doi pași. Mai întâi, împărțiți cadrul în două linii:

```
<!--toc at top, chapter file on bottom-->
<FRAMESET rows="20%,80%">
  <FRAME name="toc" src="toc.htm">
  <FRAME name="chapter" src="ch1-top.htm">
</FRAMESET>
```

Apoi, fiecare fișier de capitol începe cu un set de cadre care stabilește cele două coloane. În acest mod, când cadrul din partea de sus încarcă o nouă pagină în cadrul din partea de jos, cele două coloane vechi sunt automat golite și încărcate cu cele două coloane noi:

```
<!-- ch1-top.htm, with chapter index on left, content on right -->
<FRAMESET cols="20%,80%">
  <FRAME name="index" src="ch1-index.htm">
  <FRAME name="content" src="ch1-content.htm">
</FRAMESET>
```

Conținutul fiecăruia dintre fișierele sursă din cele două seturi de cadre cere un atribut `target` ca parte a etichetei `<BASE>` în frunte, pentru ca această colecție să lucreze bine:

```
toc.htm          <BASE target="chapter">
ch1-index.htm    <BASE target="content">
```

De asemenea, fiecare fișier index de capitol necesită aceeași etichetă <BASE> în frunte pentru ca setul să lucreze bine.

Rezumat

Cadrele sunt un instrument folositor, deși controversat, pentru a clădi funcționalități de navigare și de afișare în siturile Web. Un set de cadre este într-adevăr destul de ușor de construit, începând cu un container <FRAMESET> și apoi detaliind definiția pentru fiecare <FRAME> și conținutul pe care îl menționează.

Suplimentar subdivizării ferestrei unui browser, puteți include și cadre inline prin folosirea etichetei <IFRAME>. Un cadru inline este, în esență, la fel ca alt conținut specializat, cum ar fi imagini sau obiecte, și este inserat și controlat similar.

Folosirea atributului `target` în paginile cu conținut vă oferă abilitatea de a lăsa un cadru să controleze cuprinsul altui cadru. Această caracteristică este foarte folositoare pentru crearea de cadre imbricate, care furnizează niveluri descrescătoare de detaliu de navigație.

Acum, după ce ați văzut modurile oficiale de a controla aspectul paginii Web, începând cu capitolul 17, „Stil simplu cu text", și sfârșind cu acest capitol, este timpul să aruncați o privire spre câteva alte metode, începând cu foile de stil JavaScript (JSSS, JavaScript Style Sheets).

Capitolul 21
Folosirea foilor de stil JavaScript

de Rick Darnell

În acest capitol

- **JSSS contra CSS 287**
- **Crearea foilor de stil cu JSSS 288**
- **Combinarea stilurilor 296**

După explorarea multiplelor posibilități cu foile de stil în cascadă (CSS, cascading style sheets) în Capitolele 18, „Utilizarea foilor de stil în cascadă", și 19, „Foile de stil în cascadă și poziționarea elementelor", este timpul să ne îndreptăm spre un mod diferit de a defini stilul: foile de stil JavaScript (JSSS, JavaScript Style Sheets) de la Netscape.

Cu JSSS, puteți folosi o foaie de stil pentru a interoga browserul despre platforma sa, dimensiunea ferestrei sale, compatibilitatea culorilor și alte elemente înainte de a atribui valori etichetelor și claselor sale. Această abordare permite unei pagini JSSS să se autoadapteze mediului în loc de a-și asuma o abordare de tip „o mărime bună la toate".

Dacă nu sunteți familiarizat cu foile de stil, ce fac ele și cum să le includeți în paginile dumneavoastră Web, întoarceți-vă și aruncați o privire în Capitolul 18.

JSSS contra CSS

În Capitolul 18, v-ați ocupat de standardul de foi de stil în cascadă de la Consorțiul World Wide Web (W3C). Acest standard este ușor diferit de JSSS al companiei Netscape, începând cu declarația de tip.

În acest punct, trebuie să spun că Netscape acceptă atât CSS, cât și JSSS. În ambele modele de stil, fiecare atribut de stil se asociază unei proprietăți, deși este reprezentat puțin diferit. De exemplu, `text-align` din CSS este același cu `textAlign` din JSSS. Ca regulă, un cuvânt cu cratimă din CSS devine un singur cuvânt în JSSS, cu inițiala celui de al doilea cuvânt transformată în majusculă.

SUGESTIE

JavaScript face distincție între majuscule și literele mici, așa că este important să țineți minte cum să convertiți valorile de două cuvinte într-un singur cuvânt, pentru a fi sigur că stilurile dumneavoastră funcționează.

O altă diferență importantă apare când aveți nevoie să marcați foaia de stil cu eticheta <STYLE>. Acolo unde CSS folosește `text/css`, JSSS folosește `text/javascript`. De exemplu, pentru a lega la o foaie de stil JavaScript externă, folosiți această etichetă:

```
<LINK REL=stylesheet type="text/javascript" href="/styles/mod.js">
```

Deși foile de stil în cascadă lucrează atât în Netscape Navigator, cât și în Microsoft Internet Explorer, foile de stil JavaScript lucrează în mod curent exclusiv cu Navigator 4.0. În ciuda diferențelor lor, JSSS și CSS au o trăsătură comună importantă: fiecare atribut de stil din CSS se reflectă și în JavaScript. Singura diferență este sintaxa folosită pentru a defini stilul.

OBSERVAȚIE

JSSS a fost propus consorțiului W3C de Netscape ca un alt posibil standard pentru foile de stil (WD-jsss-960822). A fost înregistrat și nu se știe dacă va fi pus în practică.

Ar putea ajuta să menționez, în acest punct, ce este JavaScript. JavaScript este o extensie de scriptare pentru HTML, care îmbunătățește abilitatea dumneavoastră de a controla pagina HTML și răspunde evenimentelor de utilizator fără necesitatea comunicației client-server sau a unei scriptări CGI. JavaScript este mult mai apropiat de un limbaj de programare decât de etichetele HTML. El nu poate exista, totuși, în afara HTML. Pentru a funcționa, JavaScript trebuie inclus ca parte a unei pagini Web într-un browser care înțelege JavaScript. Totuși, deși Microsoft recunoaște JavaScript prin propria sa implementare, JScript, nu recunoaște sintaxa de stil JSSS. Nimeni nu a spus vreodată că lumea Internetului este un loc comod în care să prosperi.

Netscape a dezvoltat JavaScript cu Java de la Sun. JavaScript nu este o versiune redusă de Java pentru programatorii începători. Deși este înrudit cu Java, JavaScript oferă o soluție pentru scriptare de partea clientului pentru utilizatorii cu mașini de mare putere care sunt copleșiți de comunicația client-server. Puteți citi ceva mai mult despre JavaScript în Capitolul 23, „Scurtă privire asupra limbajului JavaScript".

Crearea foilor de stil cu JSSS

După introducerea în JavaScript, este logic să începem cu obiectele care reprezintă stilul în pagina dumneavoastră Web. Netscape 4.0 introduce trei noi obiecte pentru schimbarea stilurilor.

Obiecte JSSS

Trei obiecte folosite cu JSSS reprezintă cele trei părți fundamentale ale stilului: etichete, clase și ID-urile. Vi le veți reaminti din Capitolul 18.

- **Etichete:** Elementele de marcare efectivă folosite în HTML, etichetele sunt copiii obiectului `document`, astfel că sintaxa pentru a le accesa este `document.tags.`*`tagname`*, în care *`tagname`* este una dintre etichetele HTML valide, precum `H1`, `STRONG` sau `A`.
- **Clase:** Aceste obiecte sunt o diviziune următoare a etichetelor – de exemplu, versiunile `left` și `right` ale unei etichete `<P>`. Este important să nu confundați clasele de stil cu clasele Java, care sunt blocurile de construcție ale codului Java.

Clasele sunt, de asemenea, copii pentru `document`, astfel că sintaxa lor este `document.classes.`*`classname`*`[.`*`tagname`*`]`, în care *`classname`* este identificatorul pentru clasa dumneavoastră, iar *`tagname`* este un identificator opțional pentru eticheta specifică la care se aplică. O valoare specială pentru *`tagname`* este `all`, care arată că acea clasă poate fi folosită cu oricare etichetă din pagină.

- **ID-urile:** Sunt identificatorii unici pentru fiecare etichetă individuală. Gândiți ID-urile ca nume pentru fiecare etichetă – primul `<H1>` ar putea fi „Ed" (`<H1 id="Ed">`), al doilea `<H1>` ar putea fi „Bob" (`<H1 id="Bob">`) și așa mai departe.

ID-ul este, de asemenea, un copil pentru `document` și folosește sintaxa `document.ids.`*`idname`*, în care *`idname`* este numele atributului `id`. În primul exemplu precedent, acesta este `document.isd.Ed`.

Folosind toate aceste trei noi obiecte JavaScript, puteți stabili valoarea oricărui atribut de stil din document. Obiecția la toate acestea este că puteți modifica JSSS doar când pagina se

încarcă. După ce browserul a terminat de analizat pagina, stilurile sunt stabilite – nu puteți merge înapoi cu Netscape Navigator 4.0 și să schimbați atributele, așa cum puteți face cu modelul Microsoft Internet Explorer 4.0.

În plus față de aceste trei obiecte, există, de asemenea, o metodă și o proprietate utilizată cu JSSS pe care veți dori să o folosiți.

Metodă și proprietate JSSS

Lumea JSSS oferă o metodă de ajutor în stabilirea stilurilor contextuale și o proprietate folositoare pentru atribuirea stilurilor condiționale. *Stilurile condiționale* sunt aplicate dependent de starea altor variabile, fie că ele sunt alte stiluri, fie că sunt valori într-un formular. De exemplu, ați putea verifica existența unei combinații de culori care sunt greu de citit (precum roșu pe albastru) și să schimbți culorile în variante mai plăcute.

Metoda `contextual` vă permite să specificați stilurile bazându-vă pe relația lor cu poziția altor etichete. Scopul său este același ca al sintaxei contextuale pentru CSS. Puteți aduce toate etichetele `<P>` de sub o etichetă `<H1>` la un tip de caracter de 14 puncte și toate etichetele `<P>` de sub o etichetă `<H2>` la un tip de caracter de 12 puncte. Acest mod de folosire este aplicat bazându-ne pe relația paragrafului cu etichetele de titlu, nu pe clase.

Exemplul precedent este reprezentat astfel:

```
<style type="text/javascript">
contextual(tags.h1, tags.p).fontSize = "14pt";
contextual(tags.h2, tags.p).fontSize = "12pt";
</style>
```

SUGESTIE

Veți observa că, deși am vorbit despre noile obiecte ca despre copii ai obiectului `document`, nu am folosit `document` în definițiile de stil. De ce nu?

Deoarece `document` este unul dintre documentele de nivel înalt pentru o pagină Web, folosirea sa este subînțeleasă. Îl puteți folosi pentru claritate, dar nu se cere.

Proprietatea `apply` este folosită ca o prescurtare pentru a stabili stilul unei etichete sau un grup de stiluri utilizând o funcție. De exemplu, să considerăm funcția următoare, care stabilește o culoare de fond pentru o etichetă nespecificată bazându-se pe valoarea culorii sale:

```
function setBackground(thisTag) {
  with (thisTag) {
  if (color == "yellow") {
    backgroundColor = "blue"; }
  else
    if (color == "white") {
      backgroundColor = "black"; }
  }
}
```

Nu se schimbă nimic. Toată prelucrarea are loc atunci când proprietatea `apply` este adresată în timpul procesării de stil.

Următorul fragment de cod stabilește culoarea pentru trei etichete de titlu:

```
with (tags) {
  H1.color = "white";
  H2.color = "yellow";
  H3.color = "purple";
```

```
}
tags.H1.apply = setBackground(tags.H1);
tags.H2.apply = setBackground(tags.H2);
tags.H3.apply = setBackground(tags.H3);
```

Din acest punct, se ia fiecare dintre etichete și i se aplică regulile stabilite înainte în funcție. Listingul 21.1 arată pagina completă; se stabilește mai întâi culoarea textului, apoi fundalul este schimbat prin folosirea metodei `setBackground` cu proprietatea `apply`.

Listingul 21.1. Folosirea proprietății `apply` într-o definiție JSSS.

```
<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 4.0//EN">
<HTML>

<HEAD>
<TITLE>Apply and JSSS</TITLE>
</HEAD>

<STYLE type="text/javascript">
<!--
function setBackground(thisTag) {
  with (thisTag) {
     if (color == "yellow") {
        backgroundColor = "blue"; }
     else
        if (color == "white") {
           backgroundColor = "black"; }
  }
}
with (tags) {
  H1.color = "white";
  H2.color = "yellow";
  H3.color = "purple";
}
tags.H1.apply = setBackground(tags.H1);
tags.H2.apply = setBackground(tags.H2);
tags.H3.apply = setBackground(tags.H3);

//-->
</STYLE>

<BODY bgcolor="white">
<H1>This is white on black (H1)</H1>
<H2>This is yellow on blue (H2)</H2>
<H3>This is purple on the default background color (H3)</H3>
</BODY>

</HTML>
```

Când pagina este afișată (vezi Figura 21.1), fiecare titlu capătă culoarea de fundal potrivită, chiar dacă ea nu a fost stabilită explicit în timpul definirii de stil.

Aplicarea JSSS unei pagini Web

Acum, că ați aruncat o privire iscoditoare către diferitele obiecte folosite pentru a reprezenta stilul în JavaScript, este timpul să punem realmente JSSS la treabă într-o pagină Web adevărată. Așa cum ați văzut când am vorbit de obiecte, crearea unei pagini cu JSSS este asemănătoare creării unei pagini cu CSS, exceptând sintaxa.

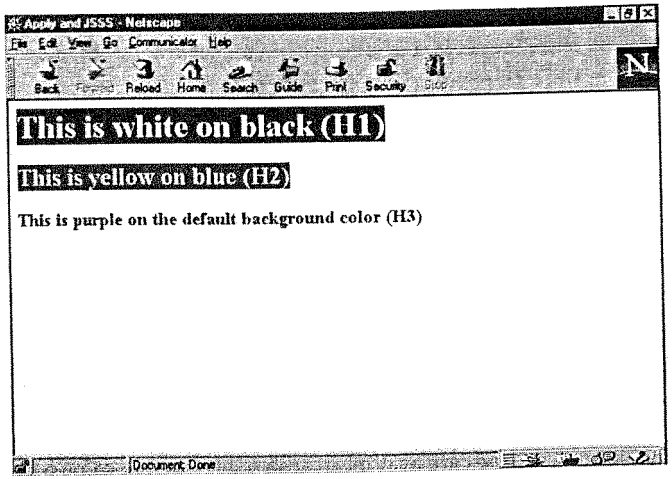


FIGURA 21.1. *Folosirea proprietății* `apply` *reprezintă un mod rapid de a crea stiluri condiționale.*

Pentru a începe, veți lucra cu o pagină construită folosind CSS (vezi Listingul 21.1 și Figura 21.2), iar apoi veți recrea stilul linie cu linie cu JSSS.

Listingul 21.2. Acesta este fișierul care utilizează CSS, pe care îl veți înlocui cu JSSS.

```
<HTML>

<HEAD>
<TITLE>A new style</TITLE>
</HEAD >

<STYLE type="text/CSS">
<!--
BODY {margin-left: 0.75in; margin-right: 0.75in; margin-top: 0.10in;
      background: white;}
H1,H2 {font-family: verdana, helvetica, arial, sans-serif}
H1 {font-size:18pt; text-align:left}
H2 {font-size:14pt; text-align:right}
P {font:garamond 12pt/11pt normal}
P.left {text-align:left}
P.right {text-align:right}
A {text-decoration: none; font-weight: bold}
-->
</STYLE >

<BODY>
<H1>More about the House of Usher</H1>
<P class="left">Shaking off from <A href="spirit.html" title="Poe's Spirit">
 my spirit</A> what must have been a dream, I scanned more narrowly the real
 aspect of the building. Its principal feature seemed to be that of an
 excessive antiquity. The discoloration of ages had been great.</P>
<H2>Not a nice place</H2>
<P class="right">Minute <A href="fungi.html" title="Fungi">fungi</A> overspread
 the whole exterior, hanging in a fine tangled web-work from the
 eaves. Yet all this was apart from any extraordinary dilapidation. No
 portion of the masonry had fallen; and there appeared to be a wild
```

(continuare)

Listingul 21.2. Continuare

```
  inconsistency between its still perfect adaptation of parts, and the
  crumbling condition of the individual stones.</P>
</BODY>

</HTML >
```

Începeți cu etichetele de stil. Containerul de bază pentru JSSS este același ca pentru CSS, cu excepția tipului și a etichetei de comentariu de închidere pentru a ascunde stilurile de browserele incompatibile:

```
<STYLE type="text/javascript">
<!-- Begin hiding from incompatible browsers

// Finish hiding from incompatible browsers -->
</STYLE>
```

Observați cele două slashuri (liniuțe oblice înclinate spre dreapta) înaintea etichetei de comentariu de închidere. În JavaScript, două slashuri marchează începutul unui comentariu. Le veți folosi pentru a restricționa JavaScript de la a privi către eticheta comentariu HTML de închidere (-->), pe care ar fi încercat să o interpreteze ca pe o instrucțiune JavaScript.

Urmează stilul corpului. Versiunea CSS originală cerea o margine la stânga și la dreapta de 0.75 inci și o margine sus de 0.10 inci. Veți face la fel cu JSSS folosind obiectul `tags.BODY`:

```
with (tags.BODY) {
  marginLeft = ".75in";
  marginRight = ".75in";
  marginTop = ".10in";
  background = "white";
}
```

Sunt stabilite cele trei valori, dar ce spuneți despre linia `with (tags.BODY)` de sus? Această caracteristică specială a limbajului JavaScript vă permite să specificați un obiect pentru un set de instrucțiuni. Orice se află între acoladele de după instrucțiunea `with` este interpretat ca și când ar începe cu `tags.BODY`.

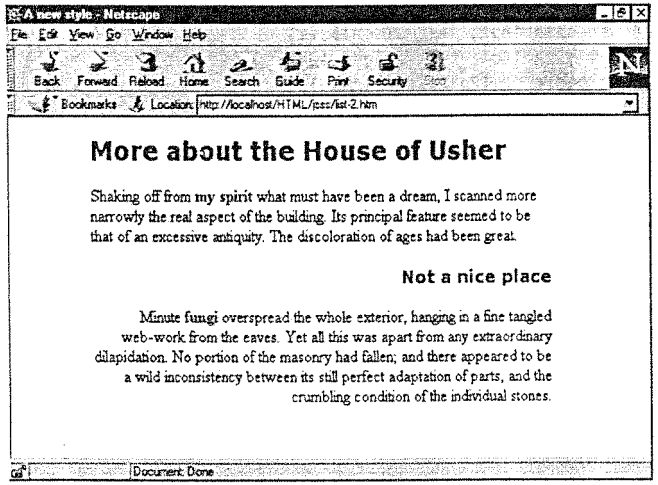


FIGURA 21.2. *Pagina construită cu stil CSS.*

OBSERVAȚIE

Anterior foilor de stil, instrucțiunea `with` era folosită în primul rând cu obiectul `Math` din JavaScript. Obiectul `Math` conține mai multe constante, precum `Pi` și `e`, și un șir de metode științifice, ca `sin` și `log`. În loc de a tasta continuu `Math.Pi`, `Math.e`, `Math.sin` sau `Math.log`, designerii pot precede o secțiune matematică cu această instrucțiune:

```
with (Math)
```

Programatorul poate urma acea instrucțiune de un set de instrucțiuni incluse în acolade. Prin această metodă, obiectul `Math` este subînțeles și se elimină o seamă de tastări și de erori de tastare.

Apoi, veți lucra asupra celor două etichete de titlu. Aceste două etichete foloseau gruparea CSS ca pe o prescurtare pentru a face definirea etichetelor mai ușoară. Din nefericire, gruparea nu este ceva ce JSSS mânuiește prea bine. Veți folosi proprietatea `apply` pentru a stabili atributele comune pentru ambele stiluri, dar rezultatul final tot implică un volum de tastări suplimentare.

```
function eval_Heading(thisTag) {
  thisTag.fontFamily = "verdana, helvetica, arial, sans-serif";
  if (thisTag == tags.H1) {
      thisTag.fontSize="18pt"; }
  else thisTag.fontSize="14pt";
}
tags.H1.apply = eval_Heading(tags.H1);
tags.H2.apply = eval_Heading(tags.H2);
```

În comparație cu același cod cerut pentru a implementa aceste stiluri cu CSS, este un pic de muncă în plus. Aceasta este bucuria de a lucra cu două forme complet diferite de sintaxă. Inconvenientele unei versiuni reprezintă punctele tari ale celeilalte și invers.

Remarcați de asemenea că nu ați stabilit încă alinierea textului la stânga și la dreapta. Voi trata acest aspect în cele ce urmează cu stilul pentru paragraf. Prima parte a lui `<P>` este clară – un stil de font cu un tip de caracter și o dimensiune:

```
tags.P.font = "garamond, serif 12 pt"
```

Acest cod reprezintă tot ceea ce aveți nevoie pentru paragraf. Veți menționa alinierea textului prin clase pe care le puteți folosi cu etichetele pentru titlu și paragraf. Ca și în CSS, puteți folosi JSSS și obiectul `classes` pentru a defini o clasă independentă de o etichetă:

```
classes.left.textAlign = "left";
classes.right.textAlign = "right";
```

Puteți folosi aceste două clase cu oricare dintre etichete pentru a forța alinierea la dreapta sau la stânga, ceea ce veți face mai târziu, când este creat cadrul HTML pentru pagină.

Ultimul stil de creat este pentru ancoră. Este o treabă simplă, în care ștergeți atributul de subliniere prestabilit și adăugați formatare aldină (bold):

```
with (tags.A) {
  textDecoration = "none";
  fontWeight = "bold";
}
```

Secțiunea de stil JSSS completată este arătată în Listingul 21.3.

Listingul 21.3. Foaia de stil JSSS completată arată întrucâtva diferit de perechea sa din CSS.

```
<STYLE type="text/javascript">
<!-- Begin hiding from incompatible browsers
```

(continuare)

Listingul 21.3. – Continuare

```
with (tags.BODY) {
  marginLeft = ".75in";
  marginRight = ".75in";
  marginTop = ".10in";
  background = "white"; }
function eval_Heading(thisTag) {
  thisTag.fontFamily = "verdana, helvetica, arial, sans-serif";
  if (thisTag == tags.H1) {
      thisTag.fontSize="18pt"; }
  else thisTag.fontSize="14pt";
}
tags.H1.apply = eval_Heading(tags.H1);
tags.H2.apply = eval_Heading(tags.H2);
with (tags.P) {
  fontFamily = "garamond, serif";
  fontSize = "12pt";
}
classes.justleft.all.textAlign = "left";
classes.justright.all.textAlign = "right";
with (tags.A) {
  textDecoration = "none";
  fontWeight = "bold";
}
// Finish hiding from incompatible browsers -->
</STYLE>
```

Când este creat corpul documentului, cum este arătat în Listingul 21.4, el se schimbă foarte puțin față de versiunea CSS. Diferența notabilă apare în cele două clase, `left` și `right`, care sunt adăugate celor două etichete de titlu pentru a stabili alinierea lor (vezi Figura 21.3).

Listingul 21.4. Corpul acestui document include adăugarea de clase etichetelor de titlu.

```
<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 4.0//EN">
<HTML>
<HEAD>
<TITLE>A new style</TITLE>
</HEAD >
<STYLE type="text/javascript">
<!-- Begin hiding from incompatible browsers
with (tags.BODY) {
  marginLeft = ".75in";
  marginRight = ".75in";
  marginTop = ".10in";
  background = "white";
}
function eval_Heading(thisTag) {
  thisTag.fontFamily = "verdana, helvetica, arial, sans-serif";
  if (thisTag == tags.H1) {
      thisTag.fontSize="18pt"; }
```



```
  else thisTag.fontSize="14pt";
}
tags.H1.apply = eval_Heading(tags.H1);
tags.H2.apply = eval_Heading(tags.H2);
with (tags.P) {
  fontFamily = "garamond, serif";
  fontSize = "12pt";
}
classes.justleft.all.textAlign = "left";
classes.justright.all.textAlign = "right";
with (tags.A) {
  textDecoration = "none";
  fontWeight = "bold";
}
// Finish hiding from incompatible browsers -->
</STYLE>
<BODY>
<H1 class="justleft">More about the House of Usher</H1>
<P class="justleft">Shaking off from <A href="spirit.html" title="Poe's Spirit">
 my spirit</A> what must have been a dream, I scanned more narrowly the real
 aspect of the building. Its principal feature seemed to be that of an
 excessive antiquity. The discoloration of ages had been great.</P>
<H2 class="justright">Not a nice place</H2>
<P class="justright">Minute <A href="fungi.html" title="Fungi">fungi</A>
 overspread the whole exterior, hanging in a fine tangled web-work from the
 eaves. Yet all this was apart from any extraordinary dilapidation. No
 portion of the masonry had fallen; and there appeared to be a wild
 inconsistency between its still perfect adaptation of parts, and the
 crumbling condition of the individual stones.</P></BODY>
</HTML >
```

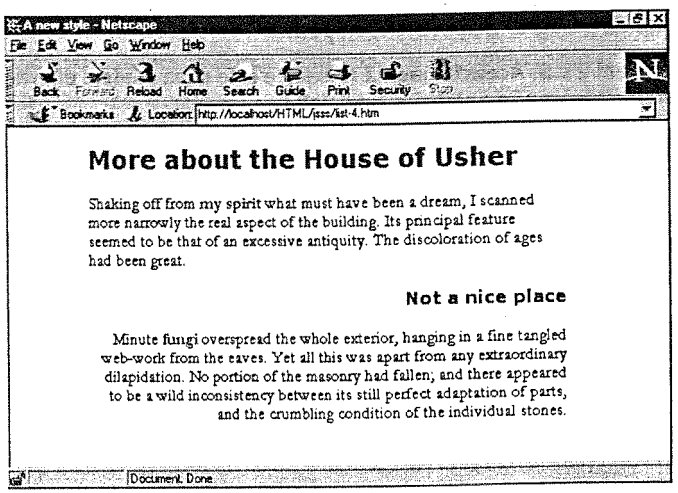


FIGURA 21.3. *Pagina JSSS cu stil terminată apare identică versiunii CSS după care a fost modelată.*

Deci ce ați dovedit prin recrearea unei pagini CSS cu JSSS? Un fapt important: sintaxa poate fi diferită, dar rezultatul este același. CSS și JSSS reprezintă căi diferite spre același rezultat.

SUGESTIE

Acum, că ați creat prima dumneavoastră foaie de stil cu JSSS, o puteți folosi într-o foaie de stil externă, similar foilor de stil legate CSS. Mai întâi, salvați secțiunea referitoare la stil a documentului în propriul său document, cu o extensie de fișier `.js`. Apoi, includeți eticheta `<LINK>` în antetul documentului pentru a aplica stilul:

```
<LINK rel="stylesheet" type="text/javascript" src="JSSS_style.js">
```

Aceasta este tot. Când pagina este încărcată, browserul va regăsi stilul și îi va aplica etichetele.

Combinarea stilurilor

În cea mai mare parte a acestui capitol, am vorbit despre CSS și JSSS ca și cum ele ar fi implementări reciproc exclusive de stiluri. Acum, că v-ați apropiat de sfârșitul capitolului, este timpul să spun adevărul: puteți folosi atât CSS, cât și JSSS pe aceeași pagină.

Cum lucrează aceasta? Lucrează ca aproape orice altceva în HTML: acela care are ultimul cuvânt câștigă. Este același mod în care foile de stil se succed în cascadă dacă includeți toate trei tipurile (legat, înglobat, inline) pe aceeași pagină. Mai întâi, browserul încarcă foaia de stil legată deoarece este primul stil pe care browserul îl întâlnește când analizează pagina.

Apoi browserul ajunge la foaia de stil înglobat. Orice configurări care intră în conflict cu stilul legat au precădere deoarece reprezintă ultima valoare pe care browserul a întâlnit-o.

Ultimul vine stilul inline, care le taie pe celelalte două. Nici unul dintre celelalte două stiluri nu are șansa de a atrage atenția browserului din nou, înainte ca textul să fie afișat, deci se aplică stilul inline.

Același proces are loc și în cazul celor două metode de creare a stilului. În loc de a include un set de etichete `<STYLE>`, includeți două. Dacă nu există nici o diferență în stiluri, ultimul stil inline va decide ce stil va fi aplicat. Pentru a vedea cum lucrează aceasta, priviți Listingul 21.5. După antet, prima foaie de stil folosește CSS pentru a reda un titlu și un paragraf de text în video invers (alb pe negru). Următorul stil folosește JSSS pentru a aduce stilul de paragraf înapoi la negru pe alb.

Listingul 21.5. Această pagină folosește cele două modele diferite de foi de stil pentru a stabili culoarea textului și a fundalului.

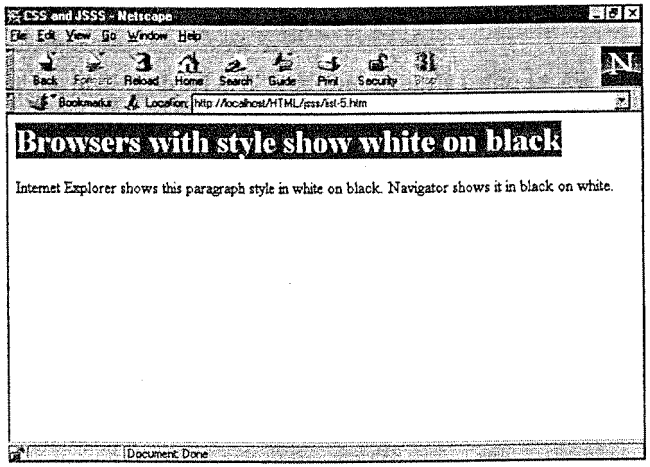
```
<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 4.0//EN">
<HTML>

<HEAD>
<TITLE>CSS and JSSS</TITLE>
</HEAD >

<STYLE type="text/css">
<!--
BODY {background: white}
H1 {color:white; background:black}
P {color:white; background:black}
-->
</STYLE >

<STYLE type="text/javascript">
<!--
```



```
with (tags.P) {
  color = "black";
  backgroundColor = "white";
}
// -->
</STYLE>

<BODY>
<H1>Browsers with style show white on black</H1>
<P>Internet Explorer shows this paragraph style in white on black. Navigator
shows it in black on white.</P>
</BODY>

</HTML>
```

Când un utilizator vizualizează pagina în Netscape Navigator, așa cum este arătată în figura 21.4, rezultatul este un titlu alb pe negru și un paragraf negru pe alb. Valoarea inițială negru pe alb pentru paragraf a fost suprascrisă de atribuirile de stil din secțiunea JSSS.

FIGURA 21.4. *Dintre două foi de stil pe aceeași pagină, ultima foaie de stil câștigă, indiferent de tipul de sintaxă utilizat în foaia de stil.*



Când utilizatorul vizualizează aceeași pagină în Internet Explorer (vezi Figura 21.5), atât titlul, cât și paragraful sunt afișate cu text alb pe negru, deoarece browserul Microsoft nu știe cum să interpreteze secțiunea JSSS. El ignoră acele valori de stil și rămâne cu versiunea CSS de foaie de stil.

Ca o regulă generală, acest tip de mixaj nu este recomandat pentru cele mai multe pagini când aplicați un stil general. Mixarea foilor de stil este folositoare dacă aveți totuși nevoie să generați un stil sau un efect pentru un browser anume. Puneți stilul prestabilit primul în document, urmat de etichetele necesare pentru a crea efectul.

Rezumat

În acest capitol, ați învățat despre foi de stil JavaScript (JSSS), un alt mod de definire a aspectului paginii. JSSS este, în esență, identic în utilizare cu foile de stil în cascadă. Sintaxa pentru JSSS este foarte diferită totuși – ca să nu spunem incompatibilă – când se folosește cu Internet Explorer. Până acum, singurul browser care acceptă JSSS este Navigator, în ciuda faptului că există un proiect care așteaptă la W3C.

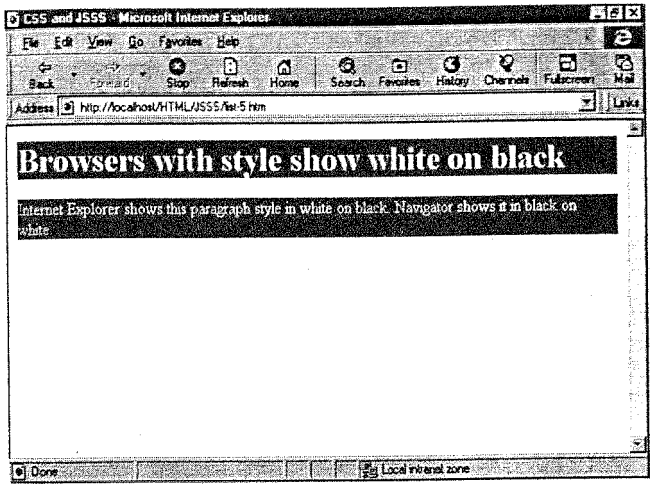


FIGURA 21.5. *Aceeași pagină vizualizată în Internet Explorer afișează titlul și paragraful în același mod, deoarece secțiunea JSSS este ignorată.*

Chiar dacă ați putea alege să utilizați CSS în loc de JSSS în paginile dumneavoastră Web pentru compatibilitate cu majoritatea browserelor, veți descoperi, probabil, că folosiți JavaScript ca limbajul dumneavoastră de scriptare pentru același motiv. Pe deasupra, CSS și JSSS pot coexista în pace pe aceeași pagină. Urmând practica browserelor Web, ultima definiție de stil întâlnită are întâietate. Puteți pune opțiunile dumneavoastră CSS prestabilite primele în linie pentru toate browserele, iar apoi să le urmați de JSSS pentru Navigator în următorul stil. Într-un browser non-JSSS, secțiunea JSSS este ignorată.

Layerele, o parte nouă a Navigator-ului, pot fi, de asemenea, prelucrate prin JSSS. Un layer este doar o secțiune a unei pagini Web – fie că reprezintă majoritatea corpului sau doar câteva linii de text – pe care o puteți poziționa, muta sau afișa independent de alte elemente din pagină. Vom trata acest subiect în capitolul următor, „Layere".

Capitolul 22
Layere

de Rick Darnell

BIBL. CENTR. UNIV. „M. EMINESCU" IAȘI INFORMATICĂ

În acest capitol

Introducere 299
Folosirea a mai mult de un layer 303
Două pagini într-una prin layere 309

Layerele (layer = strat) reprezintă modul Netscape de poziționare a elementelor HTML prin foi de stil, care poziționare este parte a unei propuneri de standard a W3C (`www.w3.org/pub/WWW/TR/WD-positioning`). Aceasta înseamnă că, la fel ca elementele propuse în poziționarea foilor de stil în cascadă, un *layer* este o secțiune a unei pagini HTML care este manevrată ca o unitate independentă. Layerele *nu sunt* o parte a standardului HTML 4.

Într-un browser compatibil cu layerele, fiecare secțiune poate fi afișată, ascunsă, mutată sau manipulată în alt fel, independent de alte secțiuni. În Netscape, layerele sunt delimitate în pagina Web cu etichetele `<LAYER>`, `<DIV>` sau `<SPAN>`, iar Microsoft suportă implementările `<DIV>` și `<SPAN>`.

SUGESTIE

Eticheta `<LAYER>` este proprietatea Netscape și nu este compatibilă cu nici un alt browser.

Ce înseamnă toate acestea? Layerele vă oferă un întreg set nou de instrumente pentru a controla poziția și vizibilitatea unor segmente întregi de pagină Web. Puteți suprapune până la zece layere și apoi să dezvăluiți pe fiecare prin rotație pentru a crea animație sau le puteți face să dispară pentru alte efecte de tranziție.

Introducere

Începeți prin a privi un exemplu simplu, arătat în Listingul 22.1 și Figura 22.1. Este un document HTML de bază care include un paragraf pentru textul de rând și un layer care conține o imagine.

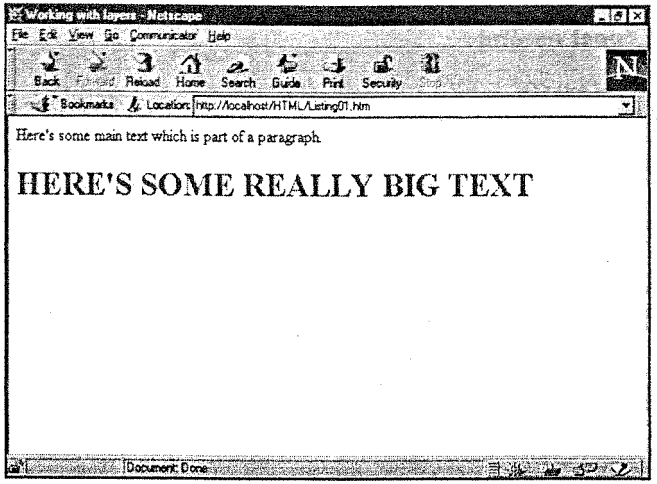
Listingul 22.1. Această pagină include un layer simplu fără elemente de control.

```
<HTML>
<HEAD>
<TITLE>Working with layers</TITLE>
</HEAD>
<BODY>
<P>Here's some main text which is part of a paragraph.</P>
<LAYER>
<H1><FONT color="red">HERE'S SOME REALLY BIG TEXT</FONT></H1>
</LAYER>
</BODY>
</HTML>
```

SUGESTIE

Veți observa lipsa declarației HTML 4 <!DOCTYPE> la începutul listingului din acest capitol. Aceasta se întâmplă deoarece layerele nu sunt parte a standardului HTML 4, deci ar fi incorect să comunicăm browserului sau analizorului SGML că documentul corespunde acestei specificații.

FIGURA 22.1. *Fără o altă formatare în acest exemplu simplu de layer, el este afișat cum ar fi oricare alt document HTML.*



Din punct de vedere al proiectării, acest layer nu este foarte interesant. Ar fi mult mai folositor dacă l-ați putea poziționa undeva în altă parte decât pe aceeași linie cu restul documentului.

Pentru a poziționa un layer, aveți nevoie de un sistem de coordonate care poate descrie orice poziție din fereastra browserului, indiferent de mărime sau formă. De aceea, toate coordonatele se consideră de la colțul din stânga-sus al ferestrei browserului.

Pentru a asigura compatibilitatea cu toate posibilele măsuri folosite în foile de stil, un layer acceptă cifre în pixeli, inci, centimetri și procente. Dacă nu este desemnat nici un sistem de măsură, layerul folosește în mod prestabilit pixeli.

Puteți vedea rezultatul poziționării bazate pe un punct de start comun în Listingul 22.2 și Figura 22.2.

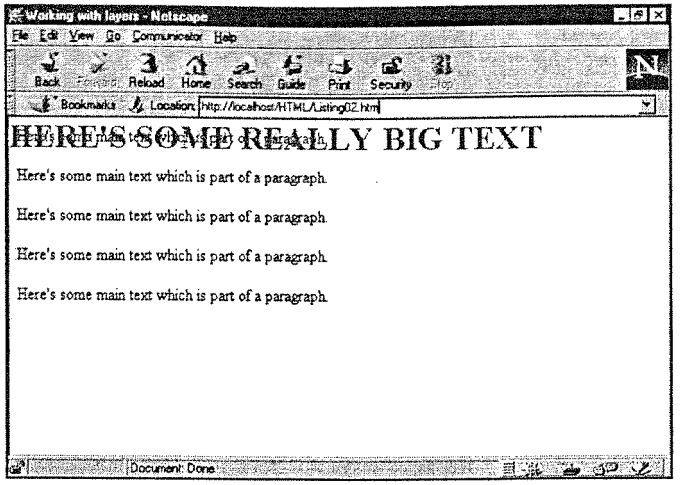
Listingul 22.2. Noua pagină include ambele elemente de text cu punct de plecare în colțul din stânga-sus.

```
<HTML>
<HEAD>
<TITLE>Working with layers</TITLE>
</HEAD>
<BODY>
<P>Here's some main text which is part of a paragraph.</P>
<P>Here's some main text which is part of a paragraph.</P>
<P>Here's some main text which is part of a paragraph.</P>
<P>Here's some main text which is part of a paragraph.</P>
<P>Here's some main text which is part of a paragraph.</P>
<LAYER top=0 left=0>
```



```
<H1><FONT color="red">HERE'S SOME REALLY BIG TEXT</FONT></H1>
</LAYER>
</BODY>
</HTML>
```

FIGURA 22.2. *Această pagină folosește poziționarea absolută pentru a plasa layerul cu imagine la același punct de start cu restul paginii.*



Iată ce se întâmplă. În primul rând, este afișat un corp de text mare, tocmai pentru a ne asigura că există din belșug în fundal când ajungem la layer. Layerul specifică o poziție absolută la colțul din stânga-sus al ferestrei browserului prin intermediul atributelor `top` și `left`. Acesta este același punct de start folosit de către toate paginile, dar, prin utilizarea celor două atribute de plasare, ați forțat textul de titlu să aibă același punct de start ca și începutul paginii.

Suplimentar plasării, puteți controla și lățimea unui layer. Listingul 22.3 este similar Listingului 22.2, dar cu două elemente de text în loc de text și grafică. Observați diferența în lățimea textului între cele două paragrafe din pagină (vezi Figura 22.3).

Listingul 22.3. Un layer nou este adăugat paginii Web pentru a controla lățimea textului de rând.

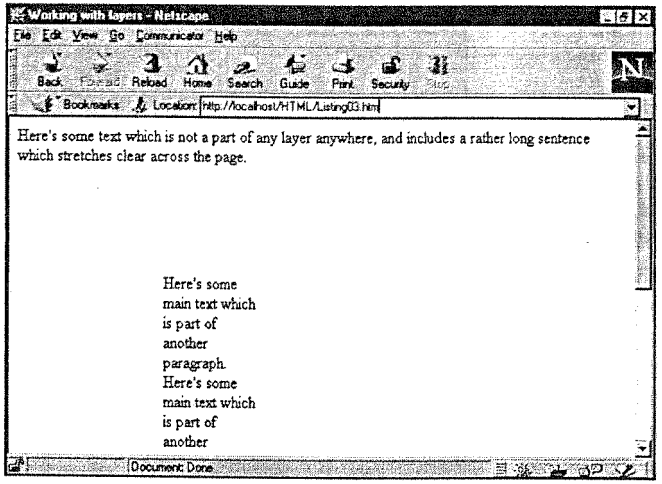
```
<HTML>
<HEAD>
<TITLE>Working with layers</TITLE>
</HEAD>
<BODY>
<P>Here's some text which is not a part of any layer anywhere, and includes
a rather long sentence which stretches clear across the page.</P>

<LAYER top=150 left=150 width=100>
<P>Here's some main text which is part of another paragraph.
Here's some main text which is part of another paragraph.
Here's some main text which is part of another paragraph.
Here's some main text which is part of another paragraph.
Here's some main text which is part of another paragraph.</P>
</LAYER>
```

(continuare)

Listingul 22.3. – Continuare

```
</BODY>
</HTML>
```

FIGURA 22.3. *Această pagină folosește controlul lățimii layerului spre a stabili marginile pentru textul din layer.*



Primul paragraf se extinde cât poate peste fereastra browserului și este împărțit conform regulilor browserului: când linia este prea lungă, se continuă pe linia următoare. Al doilea paragraf este cuprins de un layer. Poziția sa începe în josul paginii, indentat de la stânga. Apoi, lățimea sa este comprimată la 100 de pixeli, forțând apariția unei lungi coloane înguste pe mijlocul paginii.

Lucrul cu alte atribute ale etichetei <LAYER>

Ca oricare alt element din HTML, layerele au un mod de comportare prestabilit pe care ar trebui să îl luați în considerare. Mai întâi, după cum am menționat mai înainte, layerele sunt afișate în interiorul fluxului normal al documentului dacă atribute adiționale nu controlează plasarea lor.

Al doilea mod de comportare are legătură cu fundalurile și transparența. Dacă nu se specifică nici o culoare sau grafică de fundal pentru un layer, atunci el este în mod prestabilit transparent. Ați văzut acest efect ilustrat în Figura 22.2, când textul de rând se arăta prin textul titlului care era plasat peste el. Orice altă configurare pentru fundal îl face opac (vezi Listingul 22.4 și Figura 22.4).

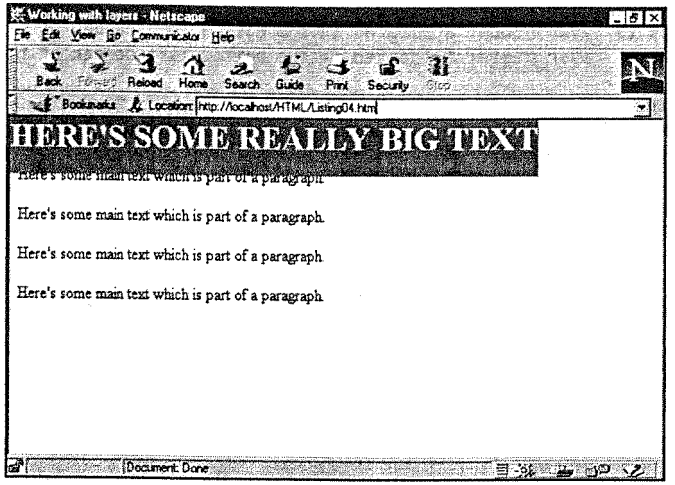
Listingul 22.4. Este adăugată o culoare de fundal layerului pentru a-l face opac.

```
<HTML>
<HEAD>
<TITLE>Working with layers</TITLE>
</HEAD>
<BODY bgcolor="white">
<P>Here's some main text which is part of a paragraph.</P>
<P>Here's some main text which is part of a paragraph.</P>
<P>Here's some main text which is part of a paragraph.</P>
<P>Here's some main text which is part of a paragraph.</P>
```



```
<P>Here's some main text which is part of a paragraph.</P>
<LAYER top=0 left=0 bgcolor="gray">
<H1><FONT color="white">HERE'S SOME REALLY BIG TEXT</FONT></H1>
</LAYER>
</BODY>
</HTML>
```

FIGURA 22.4. *Textul de rând, negru pe alb, este ascuns de textul titlului din layer, alb pe gri.*



În această pagină, culoarea de fundal este stabilită la alb, iar apoi sunt afișate cele cinci linii ale textului de rând. Urmează layerul nostru, stabilit acum la o schemă de culoare alb pe gri. Prin configurarea unui atribut de fundal, layerul este forțat să devină opac și ascunde tot textul de dedesubtul său. Puteți obține același efect folosind atributul de fundal și specificând o adresă URL pentru o imagine grafică.

Folosirea a mai mult de un layer

Exemplele precedente s-au aplicat unui singur layer, astfel încât să puteți vedea puțin din comportamentul de bază al unui layer. Este totuși posibil să adăugați mai mult de un layer, prin adăugarea mai multor etichete pentru layer. În Listingul 22.5 și Figura 22.5, au fost adăugate patru layere cu poziționare absolută, astfel încât să puteți vedea cum se stivuiesc ele pe o pagină.

Listingul 22.5. Poziționarea a patru layere în patru poziții separate pe pagină.

```
<HTML>
<HEAD>
<TITLE>Multiple Layers</TITLE>
</HEAD>
<BODY>
<LAYER top=0 left=0 width=150>
<P>This is layer number 1 located in the top left corner of the window.
This is layer number 1 located in the top left corner of the window.
This is layer number 1 located in the top left corner of the window.</P>
</LAYER>
<LAYER top=0 left=200 width=150>
```

(continuare)

Listingul 22.5. – Continuare

```
<P>This is layer number 2 located in the top right corner of the window.
This is layer number 2 located in the top right corner of the window.
This is layer number 2 located in the top right corner of the window.</P>
</LAYER>
<LAYER top=200 left=0 width=150>
<P>This is layer number 3 located in the bottom left corner of the window.
This is layer number 3 located in the bottom left corner of the window.
This is layer number 3 located in the bottom left corner of the window.</P>
</LAYER>
<LAYER top=200 left=200 width=150>
<P>This is layer number 4 located in the bottom right corner of the window.
This is layer number 4 located in the bottom right corner of the window.
This is layer number 4 located in the bottom right corner of the window.</P>
</LAYER>
</BODY>
</HTML>
```

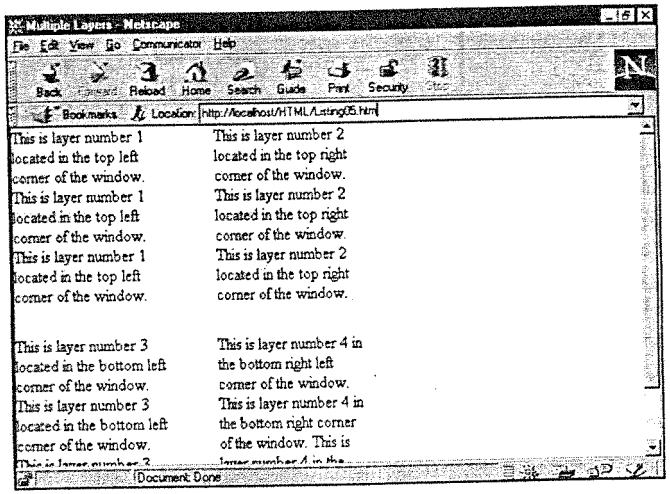


FIGURA 22.5. *În pagină sunt plasate mai multe layere, fiecare în propria sa poziție pe pagină.*

Acum, imaginația dumneavoastră trebuie să alerge sălbatică. În cele ce urmează, veți face următorul mare pas în privința layerelor: suprapunerea și schimbarea ordinii de afișare după dorință, ceea ce implică adăugarea a încă unui atribut, numit `visibility` (vezi Listingul 22.6 și Figura 22.6). Pentru a dobândi un control suplimentar asupra fiecărei secțiuni, mai trebuie să dați fiecărui layer un nume. Fiecare layer din Listingul 22.6 are deci aceleași informații despre poziție, dar include un nume și atributul `visibility`.

Listingul 22.6. Controlul stării de afișare a unui layer din stivă.

```
<HTML>
<HEAD>
<TITLE>Multiple Layers</TITLE>
</HEAD>
<BODY>
<LAYER name=one top=0 left=0 width=150 bgcolor=white>
```



```
<P>This is layer number 1.
This is layer number 1.
This is layer number 1.</P>
</LAYER>
<LAYER name=two top=0 left=0 width=150 visibility=hide below=1 bgcolor=white>
<P>This is layer number 2.
This is layer number 2.
This is layer number 2.</P>
</LAYER>
<LAYER name=three top=0 left=0 width=150 visibility=hide bgcolor=white>
<P>This is layer number 3.
This is layer number 3.
This is layer number 3.</P>
</LAYER>
<LAYER name=four top=0 left=0 width=150 visibility=hide bgcolor=white>
<P>This is layer number 4.
This is layer number 4.
This is layer number 4.</P>
</LAYER>
</BODY>
</HTML>
```

Folosirea acelorași informații de poziție și fundal pentru fiecare layer duce la acoperirea fiecărui layer, blocând vederea celor de dedesubt. Configurând layerele de la al doilea la al patrulea ca ascunse (hidden), ele sunt blocate de la vizualizare și numai primul layer va fi vizibil.

Acum, că ați configurat layerele astfel încât numai primul este vizibil, puteți atașa documentului un script pentru a controla care layer se afișează la un moment dat. Veți controla layerul cu un formular și patru butoane. Scriptul apelează o funcție care ascunde layerul curent și afișează layerul selectat, așa cum se arată în Listingul 22.7 și Figura 22.7.

Listingul 22.7. Includerea unui script pentru a afișa layerul selectat și a ascunde layerul curent.

```
<HTML>
<HEAD>
<TITLE>Multiple Layers</TITLE>
</HEAD>

<STYLE type="text/javascript">
<!--
with (document.classes.cardStack) {
  all.top = 0;
  all.left = 0;
  all.width = 150;
}
//-->
</STYLE>

<SCRIPT language="javascript">
<!--
var topLayer = 0 //the initial layer on top
function showLayer( newTop ) {
  document.layers[newTop].visibility = "show";
  document.layers[topLayer].visibility = "hide";
```

(continuare)

Listingul 22.7. – Continuare

```
  topLayer = newTop;
}
function initializeLayers () {
   for (i = 1; i<document.layers.length; i++) {
      document.layers[i].visibility = "hide";
   }
   document.layers[0].visibility = "show";
   document.layers["form"].visibility = "show";
}
//-->
</SCRIPT>

<BODY onLoad="initializeLayers()">
<LAYER name="one" class="cardStack">
<P>This is layer number 1.
This is layer number 1.
This is layer number 1.</P>
</LAYER>
<LAYER name="two" class="cardStack">
<P>This is layer number 2.
This is layer number 2.
This is layer number 2.</P>
</LAYER>
<LAYER name="three" class="cardStack">
<P>This is layer number 3.
This is layer number 3.
This is layer number 3.</P>
</LAYER>
<LAYER name="four" class="cardStack">
<P>This is layer number 4.
This is layer number 4.
This is layer number 4.</P>
</LAYER>
<LAYER name="form" top=150>
<FORM>
<INPUT type="button" value="Show Layer 1" onClick=showLayer('one')">
<INPUT type="button" value="Show Layer 2" onClick=showLayer('two')">
<INPUT type="button" value="Show Layer 3" onClick=showLayer('three')">
<INPUT type="button" value="Show Layer 4" onClick=showLayer('four')">
</FORM>
</LAYER>
</BODY>
</HTML>.
```

În Figura 22.6, layerul 1 era vizibil, dar, în figura 22.7, layerul 1 este ascuns și layerul 3 este vizibil după ce se execută clic pe butonul Show Layer 3.

După cum puteți vedea, ați adăugat un pic acestei pagini pentru a o face mai pronunțat modulară. Iată ce s-a întâmplat. Mai întâi, ați alcătuit un stil generic pentru fiecare dintre layerele din stivă. Dacă voiați să schimbați poziția, culoarea textului sau alt atribut de stil, acum este mult mai ușor să schimbați stilul decât să schimbați fiecare etichetă individual.

FIGURA 22.6. *Fiecare layer este stivuit deasupra celuilalt. Deoarece toate celelalte layere sunt ascunse vederii, layerul 1 este singurul vizibil.*

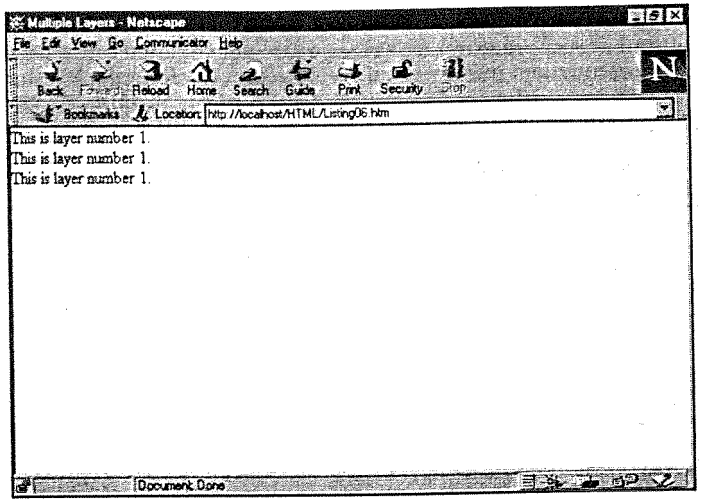
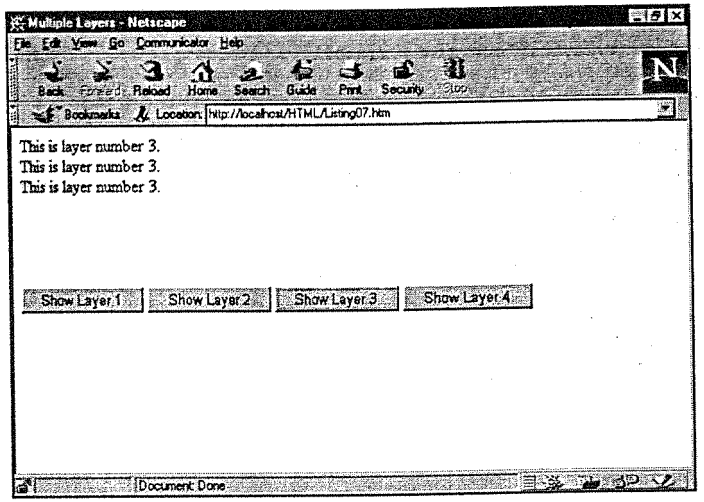


FIGURA 22.7. *Un clic pe unul dintre butoane arată layerul respectiv.*



Când pagina se încarcă, este apelat un script pentru a inițializa vizibilitatea tuturor layerelor. Acest script folosește obiectul `layers` pentru a interveni în fiecare layer și a stabili atributul său ca `hide`, apoi se întoarce pentru a stabili celelalte layere ca `show`.

După ce au loc toate acestea, utilizatorul vede ecranul inițial. Formularul este simplu, constând din patru butoane. Când se execută clic pe un buton, este apelată funcția `showLayer` cu un argument – numele layerului pe care utilizatorul dorește să îl vadă. Primele două linii ale funcției schimbă atributul `visibility` al layerului selectat în `show`, iar al layerului vizibil anterior în `hide`. Deci alegerea utilizatorului devine layerul vizibil curent.

Deplasarea unui layer în pagină

Ultimul artificiu referitor la layere pe care vi-l voi arăta este capacitatea de a deplasa un layer pretutindeni într-o pagină Web. Aceasta vă oferă prilejul câtorva efecte interesante, chiar dacă superflue, precum deschiderea cortinei sau alunecarea imaginii (vezi Listingul 22.8 și Figura 22.8).

Listingul 22.8. Prin mutarea progresivă a unui layer de-a lungul ecranului, obțineți același efect ca o deschidere de cortină care dezvăluie o scenă.

```
<HTML>
<HEAD>
<TITLE>Multiple Layers</TITLE>
</HEAD>
<SCRIPT language="javascript">
<!--
function leftMove() {
 document.layers["curtain"].moveAbove(document.layers["thetext"]);
 curtainWidth = 800;
 var moveBy = -1;
 while ( curtainWidth> -1) {
  document.layers["curtain"].offset(moveBy,0);
  curtainWidth += moveBy;
}
document.layers["curtain"].visibility = "hide";
}
//-->
</SCRIPT>
<BODY>
<LAYER name=curtain bgcolor="red" top=0 left=0>
<H1> This is a heading.</H1>
</LAYER>
<LAYER name=thetext top=5 left=5>
<P>This <A href="tempor.html" title="Eveniet">hyperlink</A>
is part of some
text. It's here so you can see what's happening with the layer moving
across the screen like the curtain on a stage.</P>
</LAYER>
<script language="javascript">
<!--
leftMove();
//-->
</SCRIPT>
</BODY>
</HTML>
```

Această pagină începe cu un titlu într-un layer. Întrucât este primul element de pe pagină, el ocupă automat poziția din stânga-sus și umple fereastra de la stânga spre dreapta. Fondul roșu face cortina opacă. Următorul layer este textul de rând, poziționat la colțul stânga-sus al ferestrei în spatele textului titlului.

SUGESTIE

Această pagină este și un exemplu de planificare pentru browserele fără layere. Dacă această pagină ar fi afișată în alt browser decât Netscape, utilizatorul ar vedea doar titlul peste textul de rând în prezentarea HTML normală a elementelor.

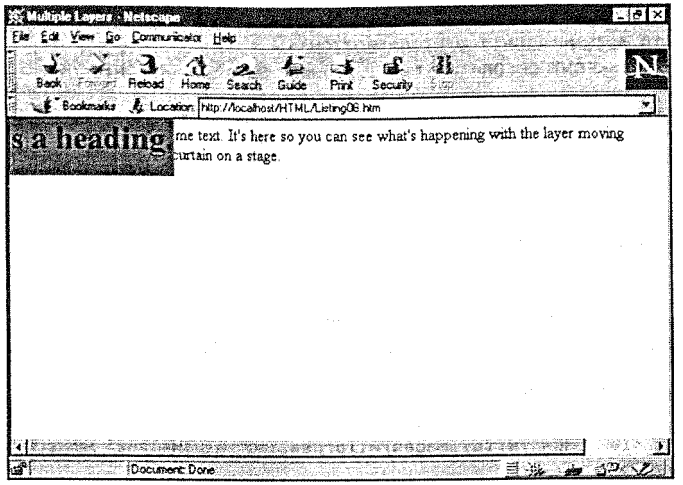


FIGURA 22.8. *Cortina cu textul de titlu este deplasată de-a lungul ecranului de la stânga la dreapta pentru a dezveli textul de rând de dedesubt.*

În continuare, aruncați o privire către script. Pentru a începe, extrageți proprietatea `windowWidth` din fereastra curentă, ceea ce vă permite să cunoașteți lățimea spațiului în care va lucra cortina. Apoi, începeți cu o valoare inițială, care este decrementată pentru a deplasa cortina într-o mișcare de la stânga la dreapta folosind metoda `moveTo` pentru layer. După ce layerul este mutat deasupra, începutul este decrementat cu 10 pixeli și bucla continuă până când ecranul este traversat.

Ca o precauție suplimentară pentru a asigura deplasarea cortinei în afară, stabiliți proprietatea `visibility` a layerului la `hide`, pentru a fi sigur că este complet ascunsă vederii în pagină.

Două pagini într-una prin layere

Ca ultimă parte a lucrului cu layere, veți vedea cum să combinați JSSS (JavaScript Style Sheets) și layerele pentru a construi două pagini HTML care sunt combinate într-una singură. Apoi, veți adăuga un formular astfel ca utilizatorul să poată alege pagina pe care dorește să o vizualizeze. Paginile combinate vor lua un pic mai mult să se încarce decât fiecare în parte, dar timpul de încărcare pentru paginile combinate este mai mic decât timpul total pentru a încărca două pagini separat de la server.

Ambele pagini sunt construite inițial dintr-un șablon folosind HTML standard. Combinați conținutul provenit de la ambele pagini prin inserarea lor în propriile layere, împreună cu o formatare suplimentară de text prin JSSS, iar apoi adăugați layerele astfel încât să se cuprindă unul pe altul.

Prima pagină este o prezentare a primejdiilor de izbucnire a unui incendiu în ținuturile sălbatice. Începeți cu corpul documentului, arătat în Listingul 22.9.

Listingul 22.9. Fundamentul paginii două-într-una include un antet, o notă de subsol și un tabel în mijlocul conținutului.

```
<BODY BGCOLOR="white">
<!-- Heading for the whole document -->
<TABLE BORDER="0" CELLSPACING="0" WIDTH="100%"
       BGCOLOR="#000000" CELLPADDING="10">
 <TR>
```

(continuare)

Listingul 22.9. – Continuare

```
  <TD ALIGN="RIGHT" VALIGN="BOTTOM">
    <H1><FONT COLOR="White">...Summer Safety...</FONT></H1>
  </TD>
  <TD ALIGN="LEFT" VALIGN="BOTTOM" WIDTH="110">
    <P><BR>
      <IMG SRC="/mrfd/Images/pointsource-posterized_92x139.jpg"
          HEIGHT="139" WIDTH="92"></P>
  </TD>
  <TD WIDTH="25"> </TD>
 </TR>
</TABLE>
<!-- Beginning of the actual content -->
<TABLE BORDER="0" CELLSPACING="15" WIDTH="100%">
 <TR>
  <TD WIDTH="25%" VALIGN="TOP">
    <H2>Living in the wildland/urban interface
     <IMG SRC="/mrfd/Images/Wildrnss.jpg" WIDTH="162" HEIGHT="122"></H2>
  </TD>
  <TD VALIGN="TOP">
    <P>Wildfire is a natural element in all ecosystems and environments, and
    Montana is no different. Our climate conditions lead to flammable ground
    cover such as grass, brush and trees. Everyone who has lived in Missoula
    County for any length of time can attest to smoky summer days, or may
    even have seen some of the larger wildfires visible from the valley. Fire
    is a part of our history and a naturally occurring element where we live.</P>
    <P>A new dimension is added to wildfires by the presence of homes. Wildfire
    quickly threatens homes and homeowners, and create a new set of issues for
    firefighters.</P>
    <P><FONT COLOR="Maroon">Missoula Rural Fire District</FONT> has firefighters
    trained for containing fires in the wildland/urban interface. But compared
    to a wildfire, they are very limited in manpower and equipment, and in the
    distances and terrain they protect. The first homes to burn in the recent
    Pattee Canyon fire were lost while firefighters were en route to the
    scene.</P>
    <P>If you're in an area which has a potential for wildfire, you can call
    for a representative from <FONT COLOR="Maroon">Missoula Rural Fire
    District</FONT> to come to your home and make
    <A HREF="/mrfd/Seasonal/Summer/Wildfire_Tips.htm">suggestions for creating
    defensible space</A> around your structures. <CITE>Defensible space</CITE>
    is an <DFN>area of reduced fuel</DFN> which gives your home greater odds of
    surviving a fire. This is a service provided free of charge to district
    homeowners. For more information, call 549-6172 or
    <A HREF="mailto:mrfd@montana.com">send us a note</A>.</P>
  </TD>
 </TR>
</TABLE>
<!-- This is the footer for the document -->
<HR>
<P ALIGN="RIGHT">
 <A HREF="/mrfd/index.htm"><STRONG>MRFD Home</STRONG>
 <IMG SRC="/mrfd/Images/cross.gif" WIDTH="113" HEIGHT="97" ALIGN="MIDDLE">
</A></P>
</BODY>
```


Prima operație este de a crea câteva stiluri pentru document spre a-l condimenta un pic. Această foaie de stil creează trei stiluri: unul pentru numele departamentului de pompieri și câte unul pentru etichetele de citare (<CITE>) și de definiție (<DFN>):

```
<STYLE type="text/javascript">
<!--
with (classes.MRFD.all) {
  color = "red";
  fontFamily = "Helvetica, sans-serif";
  fontStyle = "bold";
}
with (tags.CITE) {
  color = "green";
  fontWeight = "bold";
  fontStyle = "normal";
}
with (tags.DFN) {
  color = "green";
  fontWeight = "normal";
  fontStyle = "italic";
}
//-->
</STYLE>
```

Clasa pentru numele departamentului de pompieri este adăugată etichetei <FONT>, astfel că textul va fi evidențiat întotdeauna, chiar dacă JSSS nu este acceptat:

```
<FONT COLOR="Red" CLASS="MRFD">Missoula Rural Fire District</FONT>
```

Celelalte etichete vor fi afișate conform cu reprezentările lor prestabilite în browserele necompatibile și cu text verde aldin sau cursiv în browserele JSSS.

Următorul pas este să adăugați un layer pentru a cuprinde conținutul documentului. Mai întâi, adăugați următoarea etichetă de layer la partea de sus a conținutului și completați-o cu </LAYER> pentru închidere:

```
<LAYER name="content_0" bgcolor="gray">
```

Acest procedeu conferă layerului un nume pe care îl puteți folosi pentru trimitere la el dintr-un script și face fondul opac, astfel încât să blocheze orice altceva de dedesubtul său. Acum, trageți conținutul din documentul al doilea și adăugați-l imediat după primul, împreună cu un alt set de etichete de layer. Porțiunea cu conținut a paginii dumneavoastră apare în Listingul 22.10.

Listingul 22.10. Partea leului din pagină include cuprinsul din două pagini pus pe una și conținut în layere individuale.

```
<LAYER name="content_0" bgcolor="gray">
<TABLE BORDER="0" CELLSPACING="15" WIDTH="100%">
 <TR>
  <TD WIDTH="25%" VALIGN="TOP">
    <H2>Living in the wildland/urban interface
      <IMG SRC="/mrfd/Images/Wildrnss.jpg" WIDTH="162" HEIGHT="122"></H2>
  </TD>
  <TD VALIGN="TOP">
    <P>Wildfire is a natural element in all ecosystems and environments, and
    Montana is no different. Our climate conditions lead to flammable ground
    cover such as grass, brush and trees. Everyone who has lived in Missoula
```

(continuare)

Listingul 22.10. – Continuare

```
    County for any length of time can attest to smoky summer days, or may
    even have seen some of the larger wildfires visible from the valley. Fire
    is a part of our history and a naturally occurring element where we live.</P>
    <P>A new dimension is added to wildfires by the presence of homes. Wildfire
    quickly threatens homes and homeowners, and create a new set of issues for
    firefighters.</P>
    <P><FONT COLOR="Maroon" CLASS="MRFD">Missoula Rural Fire District</FONT>
    has firefighters trained for containing fires in the wildland/urban
    interface. But compared to a wildfire, they are very limited in manpower
    and equipment, and in the distances and terrain they protect. The first
    homes to burn in the recent Pattee Canyon fire were lost while firefighters
    were en route to the scene.</P>
    <P>If you're in an area which has a potential for wildfire, you can call
    for a representative from <FONT COLOR="Maroon" CLASS="MRFD">Missoula Rural
    Fire District</FONT> to come to your home and make
    <A HREF="/mrfd/Seasonal/Summer/Wildfire_Tips.htm">suggestions for creating
    defensible space</A> around your structures. <CITE>Defensible space</CITE>
    is an <DFN>area of reduced fuel</DFN> which gives your home greater odds of
    surviving a fire. This is a service provided free of charge to district
    homeowners. For more information, call 549-6172 or
    <A HREF="mailto:mrfd@montana.com">send us a note</A>.</P>
 </TD>
</TR>
</TABLE>
</LAYER>

<LAYER name="content_1" bgcolor="gray" visibility="hide">
<TABLE BORDER="0" CELLSPACING="15" WIDTH="100%">
 <TR>
  <TD WIDTH="25%" VALIGN="TOP">
    <H2>Wildfire...Are you prepared?
    <IMG SRC="/mrfd/Images/flashover_small_100x118.jpg" WIDTH="160" VSPACE="10">
    </H2>
  </TD>
  <TD VALIGN="TOP">
   <P>More and more people are making their homes in woodland settings -- in
   or near forests, rural areas or remote mountain sites. There, homeowners
   enjoy the beauty of the environment but face the very real danger of
   wildfire.</P>
   <P>Wildfires often begin unnoticed. They spread quickly, igniting brush,
   trees and homes. Reduce your risk by preparing now -- before wildfire
   strikes. Meet with your family to decide what to do and where to go if
   wildfires threaten your area. Follow the steps listed here to protect your
   family, home and property.</P>
  </TD>
 </TR>
 <TR>
  <TD WIDTH="25%" VALIGN="TOP">
    <HR>
    <H2>Creating defensible space...</H2>
  </TD>
  <TD VALIGN="TOP">
    <HR>
    <UL>
     <LI>Design and landscape your home with wildfire safety in mind. </LI>
```



```
     <LI>Select materials and plants that can help contain fire rather than
     fuel it. </LI>
     <LI>Use fire resistant or non-combustible materials on the roof and
     exterior structure of the dwelling. </LI>
     <LI>Treat wood or combustible material used in roofs, siding, decking
     or trim with UL-approved fire-retardant chemicals. </LI>
     <LI>Plant fire-resistant shrubs and trees. For example, hardwood trees
     are less flammable than pine or spruce. </LI>
    </UL>
   </TD>
  </TR>
  <TR>
   <TD WIDTH="25%" VALIGN="TOP">
    <HR>
    <H2>Practice wildfire safety...</H2>
   </TD>
   <TD VALIGN="TOP">
    <HR>
    <P>Find out how you can promote and practice wildfire safety around your
    home and property.</P>
    <UL>
     <LI>Contact your local fire department, health department or forestry
     office for information on fire laws. </LI>
     <LI>Make sure that fire vehicles can get to your home. </LI>
     <LI>Clearly mark all driveway entrances and display your name and
     address. </LI>
     <LI>Report hazardous conditions that could cause a wildfire. </LI>
     <LI>Teach children about fire safety. Keep matches out of reach. </LI>
     <LI>Post fire emergency telephone numbers. </LI>
     <LI>Plan several escape routes away from your home -- by car and by
     foot. </LI>
     <LI>Talk to your neighbors about wildfire safety. Plan how the
     neighborhood could work together after a wildfire. </LI>
<LI>Make a list of your neighbors' skills such as medical or
     technical. </LI>
     <LI>Consider how you could help neighbors who have special needs such
     as elderly or disabled persons. </LI>
     <LI>Make plans to take care of children who may be on their own if
     parents can't get home. </LI>
    </UL>
    </TD>
 </TR>
</TABLE>
</LAYER>
```

Acum aveți nevoie de un mod de a selecta pagina de vizualizat. Folosiți o listă derulantă într-un formular apropiat de partea de sus a paginii, care apelează o funcție pentru a afișa pagina potrivită:

```
<FORM name="topic">
Choose a topic:
<SELECT name="choice" onChange="showPage(this.selectedIndex)">
  <OPTION>The Wildland/Urban Interface</OPTION>
  <OPTION>Preparing for Wildfire</OPTION>
</SELECT>
</FORM>
```

În argumentul pentru apelul la `showPage` există două articole importante. Primul este `this`, care se referă la elementul curent din formularul curent. Al doilea este `selectedIndex`, care returnează un număr corespunzător articolului selectat. Întrucât lucrați cu calculatoare, prima opțiune este 0, a doua este 1, a treia 2 ș.a.m.d.

Ultimul pas este să creați scriptul pentru a ascunde și a afișa fiecare layer (mai multe informații despre scriptare se află în capitolul 23, „Scurtă privire asupra limbajului JavaScript"):

```
<SCRIPT language="javascript1.1">
<!--
VisibleLayer = "content_0";
function showPage ( layerIndex ) {
   chosenLayer = "content_" + layerIndex;
   document.layers[ visibleLayer ].visibility = "hide";
   document.layers[ chosenLayer ].visibility = "show";
   visibleLayer = chosenLayer;
//-->
</SCRIPT>
```

Începeți acest script prin inițializarea pointerului cu layerul vizibil curent. Apoi, funcția va accepta indexul din lista de selecție și îl va utiliza pentru a construi numele layerului selectat. Layerul curent este ascuns, layerul selectat este afișat, iar pointerul pentru layerul curent este actualizat.

Când combinați stilul, scriptul și conținutul, rezultatul este o listă derulantă cu care utilizatorii pot selecta una dintre pagini după pofta lor (vezi Figura 22.9). De asemenea, o puteți extinde cu ușurință adăugând layere și opțiuni suplimentare la formular. Dacă utilizatorii vizualizează această pagină pe un browser incompatibil, fiecare etichetă de layer este ignorată, iar combinația este văzută ca o singură pagină uriașă.

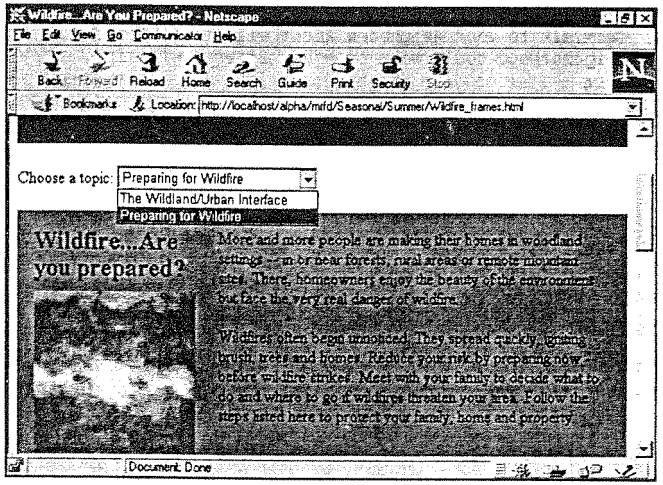


FIGURA 22.9. *Când beneficiarul alege o pagină din listă, ea este afișată automat în fereastră fără vreun efort din partea browserului.*

Mai este un ultim subiect. Ați văzut hiperlegătura din primul layer? A fost utilizată pentru a indica spre pagina care conține al doilea layer. Totuși, o puteți face compatibilă cu oricare browser. În primul rând, adăugați o ancoră imediat următoare etichetei pentru al doilea layer:

```
<A name="fire_tips"></A>
```


Apoi, modificați hiperlegătura pentru a funcționa cu scriptul dumneavoastră pe browserele compatibile și a sări la ancoră pe browserele incompatibile:

```
<A HREF="#fire_tips" onClick="showPage(0)">suggestions for creating
defensible space</A>
```

Când un utilizator execută clic pe această legătură într-un browser compatibil, subrutina de tratare a evenimentului `click` captează acțiunea utilizatorului înainte să fie făcută o tentativă de a sări la ancoră. Pe un browser incompatibil, partea `onClick` a etichetei este ignorată, iar browserul sare la ancoră.

Rezumat

Eticheta Netscape `<LAYER>` este folosită mult ca element de poziționare cu foi de stil în cascadă (discutate în capitolul 19, „Foile de stil în cascadă și poziționarea elementelor"). Ea poate fi folosită la crearea unor machete avansate în documentul dumneavoastră HTML, la crearea unor efecte de animație simplă (precum o cortină care dezvăluie conținutul) sau la combinarea paginilor într-un aranjament de diagramă.

Dată fiind asemănarea sa cu alte scheme de poziționare, W3C nu intenționează să includă eticheta `<LAYER>` ca parte a standardului HTML. Layerele rămân doar pentru Netscape Navigator 4.0, fără vreun suport dinspre Microsoft, Mosaic sau oricare alt browser.

În același timp, pentru intraneturi sau alte situri care utilizează exclusiv Netscape, layerele oferă o metodă pentru machetare HTML personalizată și efecte simple de HTML dinamic.

PARTEA a V-a

Limbaje de programare pentru HTML

Capitolul 23
Scurtă privire asupra limbajului JavaScript

de Kelly Murdock

În acest capitol

- **Ce este JavaScript? 316**
- **Cum se utilizează? 317**
- **Condiția actuală a limbajului JavaScript 318**
- **Pentru ce este bun? 318**
- **Resurse JavaScript 325**
- **Un exemplu de cod JavaScript 327**

Cu fiecare nouă versiune, HTML devine din ce în ce mai puternic, dar unele lucrări cer încă mai mult decât ceea ce pot oferi etichetele HTML de bază.

HTML este proiectat să formateze și să afișeze articole pe ecran. Însă dacă doreați ca paginile dumneavoastră Web să interacționeze cu utilizatorul, în trecut trebuia să scrieți un script CGI care să fie plasat pe server. Scripturile CGI sunt foarte capabile, dar reclamă adesea unele cunoștințe neconvenționale de programare și despre server.

O altă variantă este de a folosi JavaScript, un limbaj ușor de învățat și de folosit. JavaScript rulează pe mașina client și este înglobat în codul HTML, astfel că este ideal pentru oamenii obișnuiți cu HTML. În acest capitol, aruncați o scurtă privire asupra a ceea ce este JavaScript, învățați cum poate fi utilizat pentru a vă îmbogăți paginile Web și lucrați pe un exemplu simplu.

Ce este JavaScript?

JavaScript este un limbaj de scriptare simplu, interpretat, orientat spre obiecte, care poate fi folosit pentru a adăuga un comportament interactiv simplu unei pagini HTML prin intermediul unui script de cuvinte-cheie inserate într-o pagină Web.

JavaScript este considerat uneori un subset al limbajului de programare Java. De fapt, când priviți spre originea sa, puteți vedea că nu este adevărat. Când Sun lucra la definirea limbajului Java, Netscape crea un limbaj de scriptare pentru produsul lor LiveWire, numit LiveScript.

Între timp Java a fost lansat cu multe urale. Netscape și-a dat seama că LiveScript ar fi folositor ca mod de interfațare cu Java și a contactat Sun pentru a-i comunica ideea. Sun a ajutat Netscape să refacă LiveScript și l-a numit JavaScript. Deși posedă într-adevăr unele elemente de sintaxă similare cu cele din Java, JavaScript nu este un derivat din Java.

Este simplu

Cu JavaScript puteți îngloba scripturi direct în paginile HTML. Aceste scripturi pot fi folosite pentru a programa pagina Web să răspundă și să reacționeze la acțiunile programului de vizualizare, fără ca dumneavoastră să trebuiască să învățați un limbaj de programare complex. JavaScript ocupă un teren de mijloc între programare și marcarea HTML, ceea ce îl face accesibil fiecăruia.

OBSERVAȚIE

Cei familiarizați cu conceptele programării vor considera JavaScript foarte ușor de învățat. Totuși, ei vor constata rapid că acestuia îi lipsește puterea unui limbaj de programare complet.

Dacă nu sunteți familiarizat cu astfel de concepte, a învăța JavaScript este un pic mai complicat, dar nu imposibil. Începeți prin disecarea câtorva exemple simple. Cu HTML 4, JavaScript devine mai mult o necesitate decât un lux.

Este interpretat

Limbajele de programare cum este Java sunt *compilate* – convertite într-o formă pe care calculatorul o poate înțelege înainte de execuție. JavaScript este diferit. El este un limbaj interpretat, ceea ce înseamnă că procesul de conversie în cod de mașină are loc în timp ce rulează scriptul.

Fiind un limbaj interpretat, JavaScript este mai ușor de utilizat, dar crește și șansa de a include un script cu erori. Trebuie să verificați toate scripturile dumneavoastră înainte să le plasați pe paginile Web.

Este orientat spre obiecte

Este important că limbajul este orientat spre obiecte; aceasta înseamnă că limbajul este proiectat pentru a fi modular, cu secțiuni de cod reutilizabile, extensibile. Fiecare obiect are proprietăți, metode și subrutine de tratare a evenimentelor asociate lui. Înțelegând cum lucrează fiecare dintre aceste elemente, le puteți folosi pentru oricare obiect pe care îl întâlniți.

Proprietățile obiectului sunt, pur și simplu, atributele definitorii care întregesc obiectul. De exemplu, obiectul buton are un nume și o valoare. Acestea sunt proprietăți care ajută să definim ce este butonul.

Metode sunt acțiunile pe care le puteți exercita asupra obiectului, precum a executa clic pe un buton sau a face un obiect șir în întregime din litere mari. Metodelor le puteți transmite și parametri pentru a modifica rezultatele.

Subrutinele de tratare a evenimentelor (event handlers) sunt folosite să marcăm comenzile de executat când se întâmplă ceva. De exemplu, subrutina de tratare a evenimentului `onClick=` spune browserului ce să facă atunci când are loc execuția unui clic pe obiect.

Cum se utilizează?

Scripturile JavaScript pot fi înglobate în pagini HTML foarte ușor, folosind eticheta `<SCRIPT>`. Unul dintre atributele etichetei `<SCRIPT>`, `LANGUAGE`, este folosit pentru a specifica limbajul de scriptare la care browserul ar trebui să se aștepte. Ar trebui să arate astfel:

```
<SCRIPT LANGUAGE="JavaScript">
<!-- JavaScript code goes here -->
</SCRIPT>
```

Eticheta <SCRIPT> ar trebui să fie așezată în secțiunea <HEAD> a codului dumneavoastră HTML. Aceasta nu este o necesitate, dar ajută la crearea unui cod mai clar. Pentru secțiunile de cod JavaScript care afișează date pe ecran, este nevoie să plasați codul în secțiunea <BODY>.

OBSERVAȚIE

Există și alte limbaje de scriptare în afară de JavaScript – cel mai notabil este VBScript, care se bazează pe Visual Basic de la Microsoft. Există multe similitudini între aceste două limbaje de scriptare, dar sintaxa este diferită. Vezi capitolul 24, „Folosirea limbajului VBScript", pentru mai multe informații.

Pentru ca JavaScript să lucreze pe mașina client, el trebuie rulat pe un browser Web care înțelege JavaScript. Netscape Navigator 2.0 (sau versiunile mai noi) și Internet Explorer 3.0 (sau versiunile mai noi) înțeleg ambele JavaScript.

Pentru a vă asigura că browserele care nu înțeleg JavaScript nu încearcă să afișeze codul JavaScript, ar trebui să încadrați scriptul cu marcaje de comentariu, cum se arată în linia de cod precedentă. Aceste marcaje de comentariu ascund codul de alte browsere.

JavaScript poate fi scris direct într-un fișier HTML folosind eticheta <SCRIPT>, dar puteți și să referiți un fișier de scripturi extern prin atributul SRC, după cum se vede în continuare:

```
<SCRIPT LANGUAGE="JavaScript" SRC="http:/www.myScripts.com/coolScript.js">
```

Remarcați că fișierele JavaScript au extensia `.js`.

Condiția actuală a limbajului JavaScript

JavaScript a fost lansat ca versiune 1.0 împreună cu Netscape Navigator 2.0. Versiunea 1.1 a fost dezvoltată pentru Navigator 3.0. Acum, cu Netscape Communicator, JavaScript a fost actualizat din nou, la versiunea 1.2.

Cu fiecare nouă versiune, alte câteva obiecte au fost adăugate limbajului. În principal, aceste noi obiecte vă permit să controlați noile caracteristici din browser. Pentru o privire asupra adăugirilor la versiunea 1.2, vizitați pagina What's New in the JavaScript 1.2, la `http://developer.netscape.com/library/documentation/communicator/jsguide/js1_2.html`.

Microsoft a creat o implementare JavaScript pentru browserul său Internet Explorer numită JScript. Diferențele dintre JavaScript și JScript sunt foarte subtile, dar există. Dacă aveți în plan să creați scripturi JavaScript pentru Internet Explorer, verificați pagina de informații despre JScript creată de Microsoft la `http://www.microsoft.com/jscript`.

ATENȚIE!

Unele funcții JavaScript sunt incompatibile cu Internet Explorer. Dacă scrieți scripturi pentru ambele browsere, aveți grijă să verificați codul și pe unul și pe celălalt.

Pentru ce este bun?

Acum, că înțelegeți ce este limbajul și de unde vine, îmi imaginez că principala dumneavoastră întrebare este „Pentru ce este bun?". Puteți găsi răspunsul la această întrebare aruncând o scurtă privire către tipurile de funcții de scriptare posibile.

Validarea formularelor

Elementele de control al formularelor sunt comune pretutindeni în Web. Ele sunt implementate ușor folosind etichete HTML standard. Totuși, controlul datelor introduse în formulare poate fi dificil. În mod tradițional, acesta se făcea prin intermediul unui script CGI rezident pe server. Pe măsură ce datele erau introduse, scriptul CGI interoga serverul pentru a verifica dacă ele sunt valide. Dacă nu erau, atunci serverul trebuia să comunice înapoi clientului că datele n-ar fi funcționat. Apoi trebuia să fie reintroduse datele și returnate serverului. Acest proces laborios acapara resurse valoroase ale serverului.

JavaScript poate verifica intern formularul de date, din interiorul browserului, fără a bate la cap serverul. Apoi, doar dacă datele sunt valide, informația va fi trimisă serverului.

Validarea formularelor a fost una dintre primele utilizări ale limbajului JavaScript pe Web și încă rămâne foarte populară.

Răspunsul la introducere

Pe măsură ce datele sunt introduse în elementele de control al formularelor JavaScript, pagina Web poate răspunde acestei introduceri.

O utilizare comună pentru JavaScript în Web este de a crea jocuri simple, precum versiunea Mastermind găsită la `www.dreamlink.se/mastermind`. În acest joc, utilizatorul selectează culorile din casetele cu meniu derulant. Scriptul răspunde calculând variantele și afișând sugestii în care cifrele 1 reprezintă răspunsuri corecte. Figura 23.1 arată un joc în desfășurare.

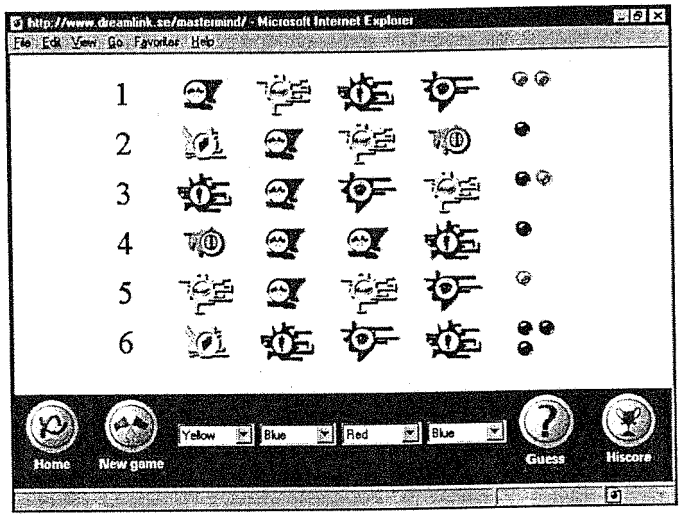


FIGURA 23.1. *O versiune JavaScript de Mastermind.*

Casete de dialog

Unul dintre modurile prin care JavaScript trimite mesaje utilizatorului este prin casete de dialog. Figura 23.2 arată concluzia jocului Mastermind printr-o casetă de dialog care anunță rezultatul victorios.

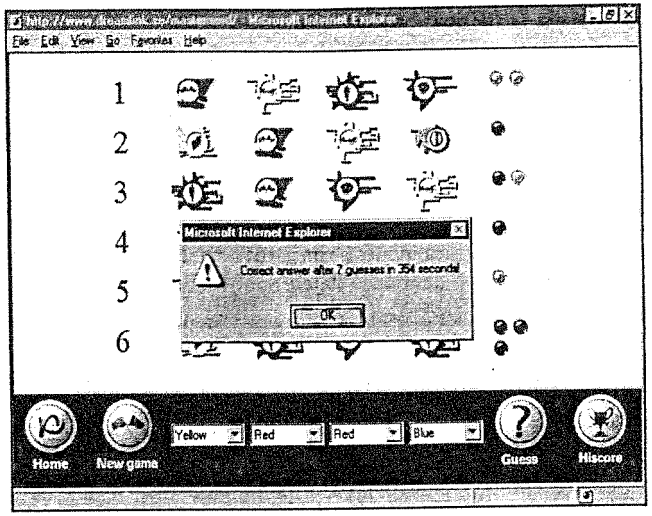


FIGURA 23.2. *Casetele de dialog afișează mesaje către spectator.*

Detectarea caracteristicilor browserului

O altă caracteristică puternică a limbajului JavaScript este capacitatea de a găsi informații despre mediul browserului. Un script dintr-un fișier HTML ar putea determina care browser este folosit de spectator și să schimbe conținutul în mod corespunzător.

Un bun exemplu de detecție a browserului este Guitar Chord Computer, găsit la `http://web.ukonline.co.uk/alastair.vance/gcc.htm`. De îndată ce accesați situl cu Internet Explorer, apare o casetă de dialog care vă informează că veți fi direcționat către o versiune care funcționează cu browserul dumneavoastră (vezi Figura 23.3).

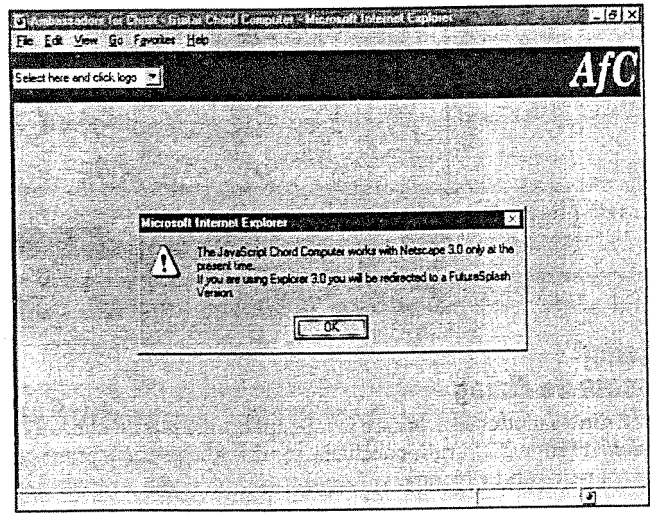


FIGURA 23.3. *JavaScript a detectat că browserul este incompatibil.*

Actualizarea proprietăților browserului

Nu numai că poate detecta ce tip de browser are spectatorul, dar JavaScript poate și schimba proprietățile browserului la cerere. Figura 23.4 arată un script care permite utilizatorilor să modifice schema de culoare din mers executând clic sau deplasând mouse-ul peste paleta de culori.

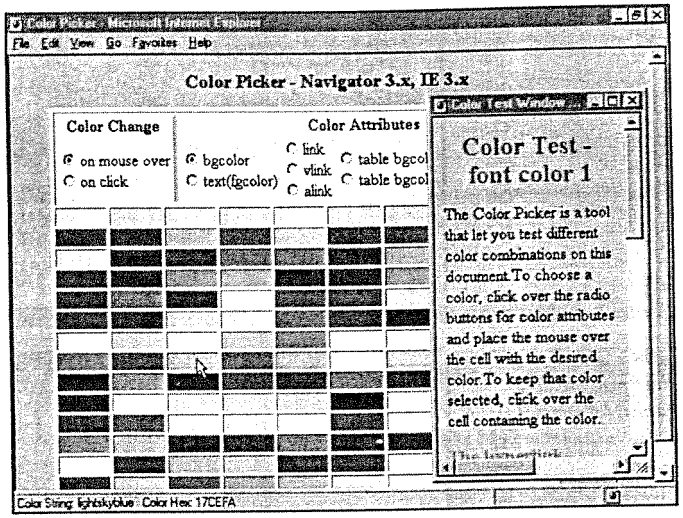


FIGURA 23.4. *Culorile browserului pot fi modificate de îndată ce utilizatorul deplasează mouse-ul peste paleta de culori.*

Acest instrument folositor poate, de asemenea, să afișeze valorile hexazecimale pentru culorile alese, ceea ce este de mare ajutor la proiectarea paginilor Web.

Capacități matematice

De asemenea, JavaScript oferă posibilități matematice complexe. Funcțiile matematice includ funcții aritmetice simple și trigonometrice complexe. Suplimentar jocurilor și actualizărilor de browser în direct, Web oferă o culegere de programe de calcul pentru a socoti orice, de la conversia temperaturii până la consumul caloric.

Suplimentar jocurilor și programelor de calcul, veți găsi o mulțime de scripturi educaționale în Web, cum ar fi îndrumarul de algebră creat de Robert Bunge, arătat în Figura 23.5. Puteți găsi acest script la `http://www.pacificrim.net/~rbunge/Algebra/Algebra.html`

Folosirea prăjiturilor pentru a păstra informații despre vizitator

Prăjiturile (cookies) sunt fișiere simple de date care păstrează informații despre vizitator. Le puteți crea și controla folosind JavaScript. Cu *prăjituri*, puteți aduna informații despre vizitatori, cum ar fi numele lor, și să le păstrați până la vizita viitoare, când îi puteți saluta personal.

Figurile 23.6 și 23.7 arată o prăjitură de uz general în acțiune. În timpul primei mele vizite la forumul JavaScript, `http://members.tripod.com/~JavaScriptForum` (Figura 23.6), scriptul detectează că sunt un vizitator nou. El înregistrează într-o *prăjitură* informațiile mele de vizitator și data la care am aruncat o privire. Apoi, în timpul celei de a doua vizite a mea, scriptul citește informațiile despre browserul meu și se uită la dată pentru a-mi transmite un mesaj personalizat.

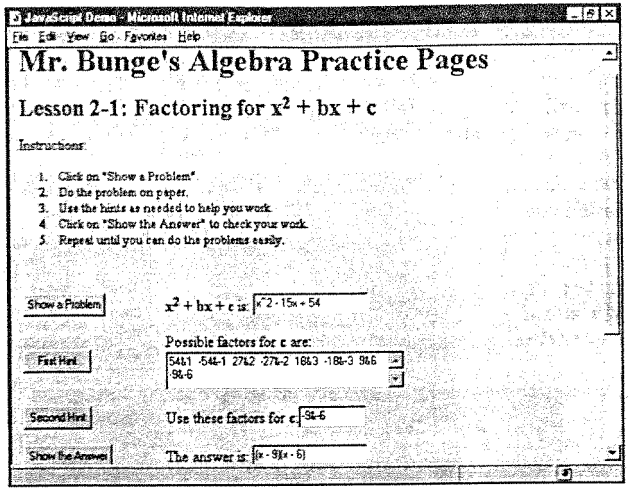


FIGURA 23.5. *Acest îndrumar de algebră beneficiază de avantajul obiectului JavaScript math.*

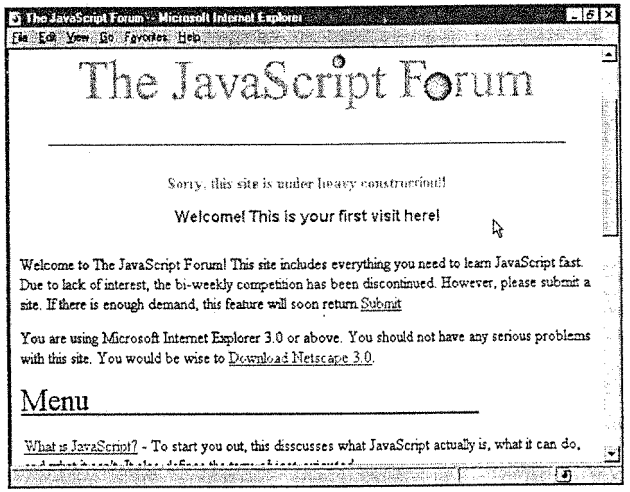


FIGURA 23.6. *Prima mea vizită la forumul JavaScript.*

Informații despre dată și oră

O altă caracteristică a limbajului JavaScript este capacitatea de a interoga sistemul asupra datei și orei. Observați marca timpului din Figura 23.7. Această caracteristică este foarte potrivită pentru a urmări ora și a fost folosită pentru a crea scripturi calendar online, ca acela arătat în Figura 23.8.

Integrarea cu Java

Deși JavaScript diferă de Java, cele două limbaje pot ușor comunica unul cu altul. Puteți include miniaplicații (applets) scrise în Java într-o pagină Web și trimite informații unui script JavaScript de pe aceeași pagină. Această metodă ajută la reducerea limitărilor limbajului JavaScript.

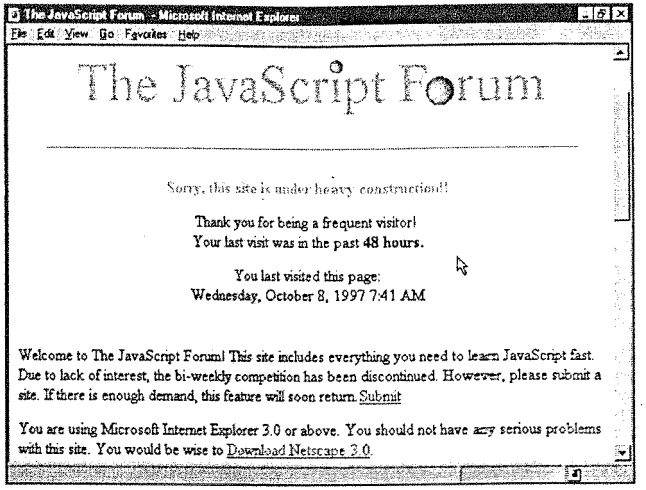


FIGURA 23.7. *O prăjitură ține evidența vizitelor mele.*

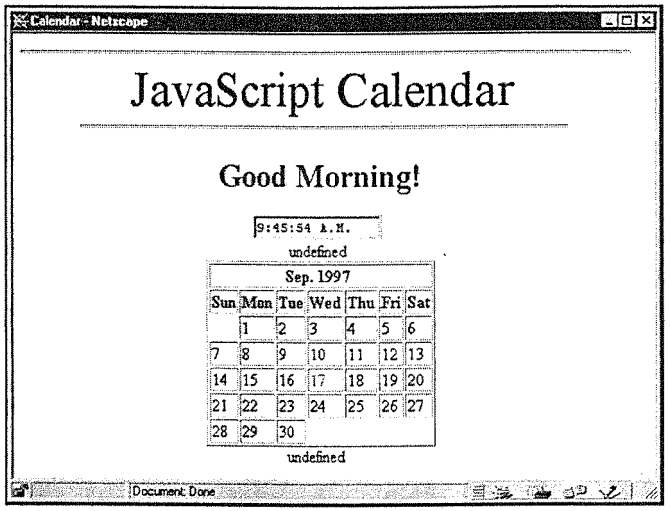


FIGURA 23.8. *Acest script calendar arată cum JavaScript poate determina date și ore.*

În exemplul ColorCenter, arătat în Figura 23.9, grila de culoare din partea de jos este, de fapt, o miniaplicație Java. Miniaplicația transmite valoarea de culoare selectată scriptului, care afișează acea culoare împreună cu valoarea sa hexazecimală.

OBSERVAȚIE

JavaScript poate lucra și cu alte tehnologii. Poate fi folosită, de exemplu, pentru a crea comportamente și mișcări simple pentru lumile VRML.

Grafică elementară

Pentru funcții grafice avansate, aveți într-adevăr nevoie de Java, dar pentru un control grafic simplu, precum afișarea de imagini bitmap sau a unor linii de desenare, JavaScript este suficient.

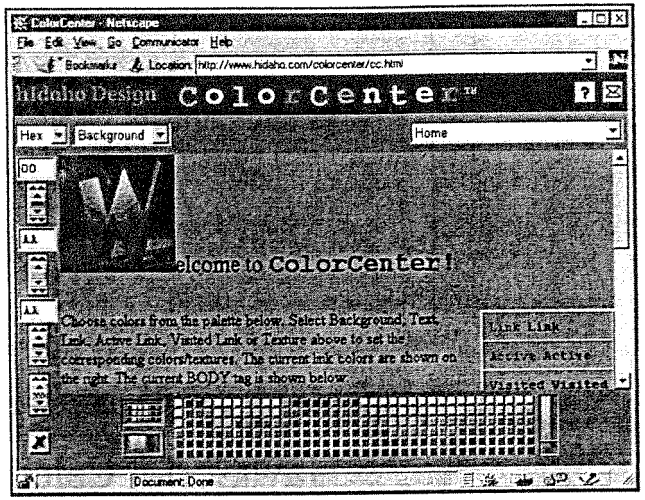


FIGURA 23.9. *ColorCenter arată cum miniaplicațiile Java și JavaScript pot fi utilizate într-o singură pagină Web.*

DoodlePad, proiectat de Bill Dortch, este creat în întregime în JavaScript. El poate desena forme și linii simple (vezi Figura 23.10).

Aceste scripturi simple din figurile precedente sunt doar o minimostră de cod JavaScript dintre multele care pot fi găsite online. În secțiunea de resurse care urmează, veți găsi multe arhive pline cu coduri JavaScript.

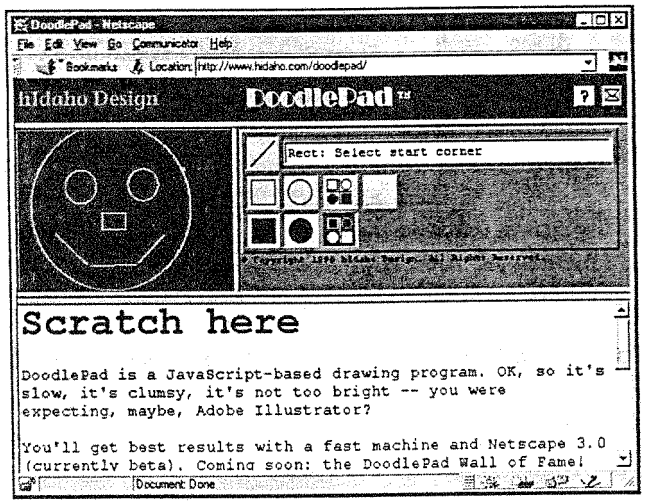


FIGURA 23.10. *DoodlePad prezintă caracteristicile de desenare simplă din JavaScript.*

HTML dinamic

În ultima versiune de HTML, HTML 4, scriptarea joacă un rol critic în controlul aspectului paginilor Web. HTML dinamic folosește puterea scriptării pentru a adăuga funcționalitate, precum poziționarea absolută și răspunsul la evenimente. Veți găsi mai multe informații despre JavaScript în Capitolele 28, „Lucrul cu utilizatorul – Evenimente", și 29, „HTML și scripturi – Combinația DHTML".

Resurse JavaScript

Există multe resurse în Web pentru a învăța și a experimenta JavaScript. Una dintre cele mai bune compilații de exemple JavaScript poate fi găsită la Gamelan, `http://javascript.developer.com`. (vezi Figura 23.11).

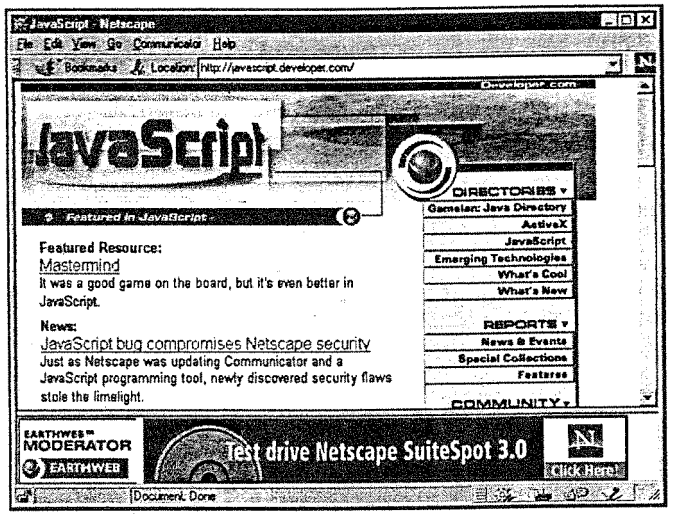


FIGURA 23.11. *Situl Gamelan este o sursă importantă de informații de programare pentru Web.*

Un alt sit bun este JavaScript World. la `www.jsworld.com`. Acest sit include o arhivă vastă, precum știri și grupuri de discuții (vezi Figura 23.12).

Multe situri JavaScript de pe Web oferă scripturi de uz general, pe care le puteți include în paginile dumneavoastră proprii. Ele pot fi foarte potrivite și nu cer deloc programare. Un astfel de sit este Cut-N-Paste JavaScript (vezi Figura 23.13), `www.infohiway.com/javascript`. Scripturile lor se întind de la scripturi simple până la text cu defilare, programe de calcul complexe și prezentări grafice gen diapozitive.

În plus față de aceste situri, Netscape posedă câteva bune posibilități de informare asupra studiului și folosirii JavaScript.

Specificația oficială a limbajului JavaScript 1.1 poate fi găsită la `http://home.netscape.com/eng/javascript/index.html`. Adăugirile la specificația 1.2 se găsesc la `http://developer.netscape.com/library/documentation/communicator/jsguide/js1_2.html`.

Netscape posedă și un ghid de creație, care este de neprețuit pentru autorii de scripturi care tocmai pornesc la drum. Îl puteți găsi la `http://home.netscape.com/eng/mozilla/3.0/handbook/javascript/index.html`.

FIGURA 23.12. *JavaScript World este o altă resursă de calitate.*

FIGURA 23.13. *Cut-N-Paste JavaScript oferă un mod lejer de a pătrunde în lumea scriptărilor.*

În sfârșit, DevEdge Newsgroup FAQ on JavaScript poate fi găsit la: `http://developer.netscape.com/support/faqs/champions/javascript.html`.

Instrumente JavaScript

Pentru a ajuta dezvoltatorii în lucrările lor de scriptare, Netscape a creat mai multe instrumente care pot face crearea de scripturi mai ușoară.

Netscape JavaScript Debugger, găsit la `http://search.netscape.com/eng/Tools/JSDebugger/relnotes/relnotespr1.html`, poate ajuta la depistarea erorilor de programare, permițând utilizatorilor să umble prin cod pas cu pas și să examineze valorile.

Adițional programului de depanare, Netscape lucrează la un mediu vizual de dezvoltare pentru JavaScript. Acest instrument puternic vă va ajuta să creați scripturi prin intermediul unui editor de machete cu posibilități de tragere și plasare (drag-and-drop). Puteți găsi ce este mai nou în Visual JavaScript 1.0 la `http://search.netscape.com/eng/Tools/VisualJS/relnotes/relnotespr2.html`.

Acum, că am tratat resursele disponibile și am pomenit câteva instrumente de ajutor, aruncați o privire rapidă asupra codării în JavaScript printr-un exemplu simplu.

Un exemplu de cod JavaScript

Am vrut ca acest capitol să fie o prezentare succintă a limbajului JavaScript și a posibilităților sale, fără a intra în detalii. Totuși, înainte de a termina, veți arunca o scurtă privire asupra unui exemplu de lucru cu limbajul.

Mai întâi, este necesar să creați o structură HTML. Codul JavaScript va fi inserat unde este scris `place code here`:

```
<HTML>
<HEAD>
<TITLE>JavaScript Example</TITLE>
<SCRIPT LANGUAGE="JavaScript">
<!-- this code is only for JavaScript-capable browsers
//place code here.
// end of JavaScript code -->
</SCRIPT>
</HEAD>
<BODY>
</BODY>
</HTML>
```

După ce este creată structura pentru fișierul HTML, prima comandă JavaScript pe care o doriți este să dispuneți ca vizitatorii să își introducă numele. Obțineți această informație folosind caseta de dialog prompt. Valoarea introdusă este stocată în variabila `Name`, așa cum se arată mai jos:

```
Name = prompt("Enter Your Name:","Name");
```

Puteți, apoi, să reapelați valoarea pentru `Name` și să o introduceți pe ecran cu instrucțiunea `document.write`:

```
document.write("<H1>" + Name + "'s Litigation Simulation</H1><HR>");
```

Ați fi putut folosi un cod HTML normal spre a afișa titlul exemplului, dar, prin folosirea limbajului JavaScript puteți include numele utilizatorului pentru a personaliza pagina.

Ulterior mai trebuie adăugat cod în eticheta `<SCRIPT>`, dar, deocamdată, deplasați-vă la eticheta `<BODY>`, unde veți folosi elemente de formular HTML pentru a crea unele casete de text.

Mai întâi, includeți unele instrucțiuni pentru respectiva activitate:

```
For this simulation, you start out with $500,000 dollars. Read the news
➥and select one of the buttons below. The fees or rewards will
➥automatically be added or deducted from your money.
```

Următorul pas este să creați trei câmpuri de formular noi: unul care arată banii dumneavoastră, unul care arată banii avocatului dumneavoastră și unul pentru știri:

```
<FORM METHOD=POST>
Your Money (in hundred thousands): <INPUT TYPE=text NAME="litsmoney"
➥VALUE="5">
```

```
<BR>
Your Lawyer's Money (in hundred thousands): <INPUT TYPE=text
➥NAME="lawsmoney" VALUE="0">
<BR>
News: <INPUT TYPE=text NAME="news" SIZE=50 VALUE="You were bit by
➥the neighbor's dog">
<BR><HR>
```

Atributele NAME sunt nume variabile pe care JavaScript le poate utiliza pentru a controla valorile acestor câmpuri.

Pe urmă, aveți nevoie de trei butoane de acționare. Ele sunt create folosind sintaxa de formular HTML standard. Comanda onClick= este o subrutină de tratare a evenimentelor care transmite codului unde să meargă când se execută clic pe buton. Încheiați această secțiune cu eticheta </FORM>:

```
<INPUT TYPE=button NAME="settle" VALUE="Settle Out of Court"
➥onClick="payment1(this.form)">
<INPUT TYPE=button NAME="sue" VALUE="Take it to Court"
➥onClick="payment2(this.form)">
<INPUT TYPE=button NAME="drop" VALUE="Drop Case"
➥onClick="payment3(this.form)">
<BR>
</FORM>
```

Payment1, payment2 și payment3 sunt funcții pe care le veți include în eticheta <SCRIPT> care modifică valoarea câmpului când butoanele sunt acționate.

Fiecare dintre aceste funcții este destul de asemănătoare celorlalte. Cuvântul cheie form este transmis ca parametru, astfel ca funcția să poată avea acces să modifice valoarea formularului. Fiecare funcție payment modifică valoarea câmpurilor de bani în mod diferit:

```
function payment1(form){
    form.litsmoney.value++;
    form.lawsmoney.value++;
    newnews(form,index);
    index++;
}
function payment2(form){
    winorlose = Math.abs(Math.sin(form.lawsmoney.value));
    if (winorlose > 0.5)
      form.litsmoney.value++;
    else
      form.litsmoney.value--;
    form.lawsmoney.value++;
    newnews(form,index);
    index++;
}
function payment3(form){
    form.litsmoney.value--;
    form.lawsmoney.value++;
    newnews(form,index);
    index++;
}
```

Observați că funcția payment2 include o serie de funcții matematice pentru a calcula un rezultat semialeator, care este salvat într-o variabilă numită winorlose. Instrucțiunile if-else sunt folosite pentru a modifica sumele de bani pe baza rezultatului din winorlose.

După ce valorile câmpului sunt modificate, este apelată o altă funcție, numită newnews, pentru a actualiza câmpul news. Ea primește parametrul form și o altă variabilă, numită index, care o ajută să se descurce cu noile intrări.

Pentru a folosi variabila index, este nevoie să o inițializați la pornirea scriptului. Cuvântul cheie var o particularizează ca variabilă, iar valoarea sa inițială este stabilită ca mai jos:

```
var index=1;
```

Toate funcțiile payment modifică variabila index prin creșterea valorii sale cu unu. Funcția newnews schimbă câmpul news funcție de valoarea lui index:

```
function newnews(form,index){
    if (index==1)
      form.news.value = "You were fired for sleeping on the job.";
    else if (index==2)
      form.news.value = "You were run over by a bus.";
    else if (index==3)
      form.news.value = "You were injured by a defective product.";
    else if (index==4)
      form.news.value = "Your publisher stole your copyright.";
    else if (index==5)
      form.news.value = "Your insurance won't cover your house fire.";
    else if (index==6)
      form.news.value = "That's all the news, you can keep going
➥if you want.";
}
```

Acesta este un exemplu simplu, dar el arată mai multe utilizări ale limbajului JavaScript. Figurile 23.14 și 23.15 arată exemplul la lucru.

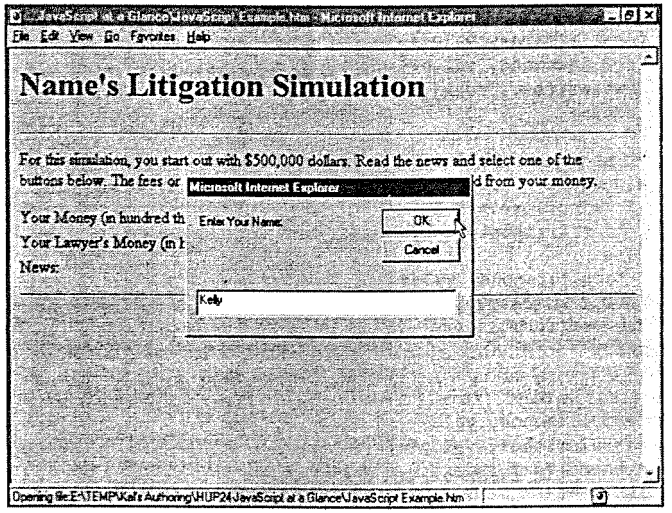


FIGURA 23.14. *O casetă de dialog din exemplu, care permite utilizatorilor să își introducă numele.*

Codul complet pentru acest exemplu este prezentat în Listingul 23.1.

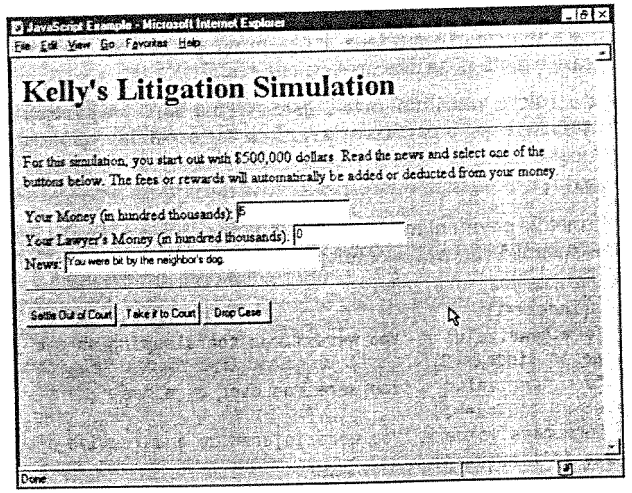


FIGURA 23.15. *Litigation Simulation este, un exemplu de cod JavaScript.*

Listingul 23.1. Exemplu de cod JavaScript pentru simularea unui litigiu.

```
<HTML>
<HEAD>
<TITLE>JavaScript Example</TITLE>
<SCRIPT LANGUAGE="JavaScript">
<!-- this code is only for JavaScript-capable browsers
Name = prompt("Enter Your Name:","Name");
document.write("<H1>" + Name + "'s Litigation Simulation</H1><HR>");
var index=1;
function payment1(form){
    form.litsmoney.value++;
    form.lawsmoney.value++;
    newnews(form,index);
    index++;
}
function payment2(form){
    winorlose = Math.abs(Math.sin(form.lawsmoney.value));
    if (winorlose > 0.5)
      form.litsmoney.value++;
    else
      form.litsmoney.value--;
    form.lawsmoney.value++;
    newnews(form,index);
    index++;
}
function payment3(form){
    form.litsmoney.value--;
    form.lawsmoney.value++;
    newnews(form,index);
    index++;
}
function newnews(form,index){
    if (index==1)
      form.news.value = "You were fired for sleeping on the job.";
```



```
    else if (index==2)
      form.news.value = "You were run over by a bus.";
    else if (index==3)
      form.news.value = "You were injured by a defective product.";
    else if (index==4)
      form.news.value = "Your publisher stole your copyright.";
    else if (index==5)
      form.news.value = "Your insurance won't cover your house fire.";
    else if (index==6)
      form.news.value = "That's all the news, you can keep going
➥if you want.";
}
// end of JavaScript code -->
</SCRIPT>
</HEAD>
<BODY>
For this simulation, you start out with $500,000 dollars. Read the news
➥and select one of the buttons below. The fees or rewards will
➥automatically be added or deducted from your money.
<FORM METHOD=POST>
Your Money (in hundred thousands): <INPUT TYPE=text NAME="litsmoney"
➥VALUE="5">
<BR>
Your Lawyer's Money (in hundred thousands): <INPUT TYPE=text
➥NAME="lawsmoney" VALUE="0">
<BR>
News: <INPUT TYPE=text NAME="news" SIZE=50 VALUE="You were bit by
➥the neighbor's dog">
<BR><HR>
<INPUT TYPE=button NAME="settle" VALUE="Settle Out of Court"
➥onClick="payment1(this.form)">
<INPUT TYPE=button NAME="sue" VALUE="Take it to Court"
➥onClick="payment2(this.form)">
<INPUT TYPE=button NAME="drop" VALUE="Drop Case"
➥onClick="payment3(this.form)">
<BR>
</FORM>
</BODY>
</HTML>.
```

Rezumat

JavaScript este doar unul dintre mai multe limbaje de scriptare care fac paginile Web interactive. Reprezintă o variantă ușoară și rapidă la limbaje de programare complexe, cum ar fi Java. În acest capitol, vi s-a oferit o scurtă privire asupra limbajului JavaScript.

Una dintre cele mai bune modalități de a învăța JavaScript este de a găsi exemple de calitate pe Web și de a le diseca. Folosirea codului altcuiva poate să vă propulseze rapid printre tehnici avansate. Ați privit mai multe scripturi eșantion culese de pe Web pentru a vedea ce tipuri de funcții sunt disponibile. De asemena, acum dispuneți de o listă de resurse care să vă ajute să vă îmbunătățiți capacitățile de scriptare.

Dacă ați investit timp în învățarea complexităților HTML-ului și credeți că paginile dumneavoastră ar putea beneficia de tipurile de caracteristici pe care JavaScript le face posibile, atunci vă îndemn să vă apucați de lucru.

Capitolul 24
Folosirea limbajului VBScript

de Robert McDaniel

În acest capitol

- **Ce este VBScript? 333**
- **VBScript și paginile dumneavoastră HTML 335**
- **Pentru ce este bun VBScript? 340**
- **Resurse pentru o informare sporită 348**

La începutul anului 1996, Netscape Communications Corporation a dat la iveală primul limbaj de scriptare de partea clientului (client-ride). Denumit la origine LiveScript, Netscape a modificat atât numele, cât și puțin din sintaxă, urmând introducerea limbajului de programare Java. Netscape a înțeles că puterea și flexibilitatea lui Java aveau unele limitări, cum ar fi execuția lentă, timpii prelungiți de încărcare a miniaplicațiilor și lipsa interacțiunii dintre Java și alte elemente Web de pe pagină. Ca urmare, Netscape a colaborat cu Sun Microsystems pentru a oferi aceleași aspect și gust de Java într-un limbaj de scriptare de categorie ușoară.

De asemena, Netscape a înțeles, în mod corect, că Java era un limbaj de programare dificil de învățat, în special pentru designerii de pagini Web cu experiență de programare redusă sau nulă. Chiar programarea de taskuri simple în Java cere abilități de departe mai presus de cele pentru codarea HTML de bază. În același timp, totuși, designerii de pagini Web doreau să aibă capacitatea de a adăuga elemente interactive, care efecuează acțiuni automat și răspund acțiunilor utilizatorului.

Pe lângă această înțelegere profundă a nevoilor designerilor de pagini Web, Netscape a sesizat și popularitatea deosebită de care se bucura Java în mass-media. Folosind un nume și o sintaxă asemănătoare, ei au putut capta o parte din frenezia mediilor de difuzare asociată cu Java. Când Netscape a lansat Navigator 2.0, ei au anunțat includerea lui JavaScript, un limbaj de scriptare ușor de învățat, de partea clientului, cu puterea și flexibilitatea unui limbaj cu caracteristici integrale de programare, dar suficient de ușor pentru a fi înglobat în codul sursă al paginilor HTML.

După ce Netscape a introdus JavaScript, primul limbaj de scriptare de partea clientului, cei de la Microsoft și-au dat seama de necesitatea înglobării unei capacități de scriptare de partea clientului în propriul său browser. Conform politicii firmei, de a prelua și a dezvolta noile idei, ei au dorit să prevadă suport pentru JavaScript în Internet Explorer. Însă, Netscape și-a înregistrat drepturile de autor atât asupra numelui JavaScript, cât și asupra implementării sale în Navigator 2.0. Prin urmare, cei de la Microsoft au fost nevoiți să creeze propria lor implementare de sintaxă JavaScript, pe care au numit-o JScript.

O dată realizată porțiunea de preluare a politicii, cei de la Microsoft au trecut la extinderea capacităților de scriptare de partea clientului. Această țintă a dus la introducerea lui VBScript 1.0 în Internet Explorer 3.0. Întrucât JScript încearcă să mențină aceeași sintaxă ca JavaScript, VBScript este al doilea limbaj de scriptare de partea clientului.

Chiar dacă VBScript este un limbaj de programare cu caracteristici integrale, el nu este des folosit de către producătorii de pagini Web, în primul rând datorită lipsei suportului

pentru el. În mod obișnuit, numai Internet Explorer de la Microsoft posedă suport nativ pentru scripturile VBScript. Și, întrucât aproape tot ce puteți face în VBScript puteți face și în JavaScript, majoritatea producătorilor aleg JavaScript drept limbajul lor de scriptare.

În concluzie, este bine sau nu să folosiți VBScript? Răspunsul depinde mult de ceea ce doriți să realizați. Dacă vreți să creați un sit Web Internet și nu doriți ca beneficiarii dumneavoastră să fie nevoiți să descarce programe plug-in, mai bine folosiți JavaScript. Însă, dacă dezvoltați un sit intranet pentru o companie care folosește exclusiv Internet Explorer, atunci VBScript este o opțiune corectă, în special dacă deja cunoașteți Visual Basic.

În acest capitol, găsiți o introducere la VBScript. Veți învăța despre următoarele:

- Ce este VBScript?
- VBScript și paginile dumneavoastră HTML
- Pentru ce este bun JavaScript?
- Resurse pentru o informare sporită

Ce este VBScript?

VBScript este o prescurtare pentru *Visual Basic Scripting Edition* (Ediția de scriptare Visual Basic). Așa cum sugerează numele său, el conține aceeași structură de limbaj și cam aceeași sintaxă ca popularul limbaj Microsoft Visual Basic. Prin crearea VBScript, Microsoft a dorit să permită producătorilor să folosească posibilitățile Visual Basic când creează scripturi de partea clientului.

Chiar dacă VBScript are rădăcini în Visual Basic, nu este nevoie să cunoașteți Visual Basic pentru a învăța VBScript. De fapt, ar trebui să fie mult mai ușor să învățați VBScript decât Visual Basic. De asemenea, pentru a folosi VBScript, nu aveți nevoie de mediul de dezvoltare și de compilator, care sunt cerute de Visual Basic.

VBScript și Visual Basic

VBScript moștenește sintaxa și structura de la limbajul de programare Microsoft Visual Basic. Totuși, există unele diferențe între cele două. Întrucât programele VBScript au fost destinate să fie înglobate în pagini Web, a fost necesar să fie făcute mai multe modificări ediției de scriptare a lui Visual Basic.

Mai întâi, deoarece VBScript este un limbaj interpretat, el rulează în mod natural mai lent decât aplicațiile Visual Basic compilate. Ca urmare, Microsoft a făcut o modificare semnificativă de limbaj pentru a face VBScript să ruleze mai rapid. Ei au transformat VBScript într-un limbaj de programare *cu tipuri libere*.

Limbajele de programare care au multe tipuri distincte de structuri de date cer să folosiți tipul de date corect în toate instrucțiunile. De exemplu, dacă într-o instrucțiune care cere tipul de date caracter se folosește un întreg, se returnează o eroare. Aceste tipuri de limbaje de programare sunt numite limbaje *cu tipuri rigide*. Limbajele care nu sunt cu tipuri rigide se spune că sunt *cu tipuri libere*.

VBScript nu este un limbaj de programare cu tipuri rigide. El conține numai o singură structură de date: variant: (varianta). Puteți stoca orice tip de date – precum logice, întregi, numere zecimale, caractere sau șiruri – în variante. Folosind o singură structură de date, Microsoft elimină în timpul execuției suprasarcina de a verifica faptul că se folosesc tipuri corecte de date.

Suplimentar eliminării oricăror alte tipuri de date din VBScript, Microsoft a trebuit să elimine și multe dintre comenzile Visual Basic care erau dependente de sistemul de

operare. Deoarece VBScript este destinat să ruleze într-un browser Web pe orice platformă, Microsoft a trebuit să se asigure că orice funcție VBScript va lucra pe toate platformele. De aceea, a eliminat orice funcții interne Visual Basic specifice unui singur sistem de operare și unei singure platforme.

Din motive de securitate, din VBScript au fost eliminate și toate comenzile Visual Basic de intrare și ieșire din fișier și cele de sistem de operare. Rularea programelor care au fost descărcate de pe Internet poate constitui un mare risc pentru buna existență a mașinii dumneavoastră. Aceste programe pot conține viruși ascunși care pot produce o grămadă de stricăciuni mașinii, de la ștergerea unor fișiere cheie până la reformatarea completă a hard-discului. Visual Basic dispune de toate comenzile de care un programator răuvoitor ar avea nevoie pentru a executa oricâte acte dăunătoare mașinii unui utilizator necunoscut.

Aplicațiile VBScript se intenționează a fi descărcate de pe situri Web de pretutindeni din Internet, astfel că o preocupare foarte reală este dacă aceste aplicații pot conține acțiuni răuvoitoare. Pentru a preveni acest fapt, Microsoft a eliminat toate comenzile Visual Basic de intrare și ieșire din fișier și cele de sistem de operare, care stau la baza programelor de virusare. Dacă o aplicație nu are acces la sistemul de fișiere sau sistemul de operare al mașinii pe care rulează, ea nu poate să aducă modificări sistemului de fișiere, cum ar fi ștergerea fișierelor, sau să invoce comenzile sistemului de operare, cum ar fi formatarea hard-discului. Prin eliminarea acestor instrucțiuni, Microsoft asigură utilizatorii că ei pot descărca și rula aplicații VBScript fără teamă de atacuri vătămătoare contra calculatoarelor lor.

O dată cu eliminarea multor comenzi Visual Basic din VBScript, Microsoft a adăugat și câteva caracteristici noi ediției de scriptare. Datorită mediului unic în care rulează VBScript, au fost necesare constante și funcții negăsite în produsul Visual Basic complet. Unele dintre aceste funcții sunt incluse pentru a reduce timpul de programare cerut de dezvoltarea aplicațiilor VBScript.

De exemplu, funcția `Split` este o funcție VBScript inexistentă în Visual Basic. Când este folosită, ea desparte în două un șir la toate aparițiile unui caracter specificat, precum virgula. Această funcție este foarte utilă pentru scripturile de partea clientului, pentru că multe date din mediu se găsesc în șiruri care sunt delimitate de unele caractere speciale. Puteți folosi funcția `Split` pentru a despărți rapid în două aceste date și a lucra cu porțiunea care prezintă interes pentru aplicația dumneavoastră.

Platforme VBScript

Prima implementare de VBScript a Microsoft a fost în edițiile beta de Internet Explorer pentru mașini cu Windows 95 și Windows NT. O dată cu ediția finală, VBScript a fost acceptat în versiunile de Internet Explorer 3.0 pentru toate platformele Windows și Power Macintosh. Microsoft lucrează cu alte companii pentru a dezvolta suport pentru VBScript în Internet Explorer pentru platforme UNIX.

NCompass, un producător terț, a creat un program plug-in Netscape, care prevede suport pentru VBScript în browserele Netscape Navigator. Acest plug-in, denumit Script-Active, este hărăzit, în primul rând, să ofere utilizatorilor Navigator-ului capacitatea de a lucra cu controale ActiveX. El oferă și un instrument de conversie, care va face paginile Web care conțin VBScript compatibile cu browserul Navigator.

Pentru a extinde utilizarea lui VBScript, Microsoft acordă licență pentru a utiliza VBScript în alte aplicații. Producătorii terți care creează alte aplicații client sau server au, acum, alternativa de a construi un interpretor VBScript în aplicația lor, permițând folosirea lui VBScript. Nu există vreo taxă de licență pentru folosirea limbajului VBScript în alte aplicații, dar termenii licenței cer imperativ menționarea Microsoft și a drepturilor de autor respective.

VBScript și paginile dumneavoastră HTML

VBScript a fost conceput ca un limbaj de scriptare de categorie ușoară, pentru a fi folosit în pagini Web. Aceasta înseamnă că limbajul de scriptare trebuie nu numai să fie înglobat într-o pagină Web, dar și să cunoască deplin acea pagină pentru a interacționa cu ea. Microsoft a beneficiat de caracteristicile elaborate anterior de Netscape pentru JavaScript, astfel că a putut prelua definițiile existente pentru a îngloba scripturi de partea clientului în pagini Web. De asemenea, a extins funcționalitatea prin definirea și disponibilizarea către script de obiecte din pagina Web și browserul Web.

Eticheta <SCRIPT>

Netscape a definit deja metoda pentru a îngloba scripturi de partea clientului în documente HTML: eticheta `<SCRIPT>`. Folosind această etichetă și asociata sa `</SCRIPT>` în HTML, browserele Web înțeleg că textul plasat între aceste etichete este cod de aplicație, nu text de document.

Eticheta `<SCRIPT>` are următoarea sintaxă:

```
<SCRIPT LANGUAGE=[scripting language]>
```

Pentru VBScript, trebuie să folosiți atributul `LANGUAGE` și să îl stabiliți egal cu `VBSCRIPT`, cum se arată mai jos:

```
<SCRIPT LANGUAGE="VBSCRIPT">
```

ATENȚIE!

Ca opțiune prestabilită, browserul Web folosește JavaScript pentru toate scripturile de partea clientului. Dacă nu este prezent atributul `LANGUAGE`, codul aplicației este interpretat drept JavaScript. Prin urmare, pentru a utiliza limbajul VBScript în oricare dintre scripturile de partea clientului dumneavoastră, trebuie să informați explicit browserul Web că folosiți VBScript, nu JavaScript. Desigur, este de bun gust să includeți atributul `LANGUAGE` de fiecare dată când folosiți eticheta `<SCRIPT>`, chiar dacă aplicația este în cod JavaScript. Dacă se folosește atributul `LANGUAGE`, iar browserul Web care încarcă pagina nu acceptă acel limbaj de scriptare, conținutul scriptului este ignorat.

Puteți avea oricât de multe apariții ale etichetei `<SCRIPT>` în pagina dumneavoastră HTML. Puteți, de asemenea, să folosiți limbaje diferite pentru apariții diferite ale etichetei `<SCRIPT>`. Totuși, se obișnuiește să aveți o singură secțiune `<SCRIPT>` întinsă, care să conțină cea mai mare parte a codului aplicației, precum subrutine, funcții și declarații de variabilă. Folosiți alte apariții ale etichetei `<SCRIPT>` doar când aveți nevoie de o funcție îndeplinită la o anume locație din document, precum scrierea datelor în pagina HTML însăși.

Ascunderea scripturilor de partea clientului de browserele incompatibile

Cele mai multe browsere Web care se folosesc în ziua de azi acceptă unul dintre limbajele de scriptare de partea clientului. Totuși aceasta nu înseamnă oricare; unii dintre vizitatorii sitului dumneavoastră vor vedea o pagină Web denaturată dacă aveți de gând să includeți VBScript în HTML-ul dumneavoastră. Nu doriți să se întâmple astfel, întrucât doriți ca paginile dumneavoastră să fie multiplatformă, multibrowser și vizualizate de oricine.

Din fericire, când Netscape a introdus eticheta `<SCRIPT>` și a implementat-o în browserul său Navigator 2.0, i-a conferit compatibilitate cu versiunile vechi. A făcut aceasta simplu, utilizând eticheta HTML de comentarii existentă și instruind browserele mai noi cu suport

pentru scriptarea de partea clientului să ignore această apariție aparte a etichetei de comentariu HTML.

Sintaxa standard pentru o etichetă <SCRIPT> în paginile dumneavoastră HTML ar arăta astfel:

```
<SCRIPT LANGUAGE="VBScript">
<!--
            Your Code Goes Here

//-->
</SCRIPT>
```

Observați a doua linie:

```
<!--
```

Această linie este prima parte a unui comentariu HTML. Browserele care acceptă scriptarea de partea clientului ignoră totdeauna începutul unui comentariu HTML. Totuși, browserele care nu acceptă scriptarea de partea clientului consideră drept comentariu tot ceea ce întâlnesc până la partea de sfârșit a etichetei de comentarii HTML și nu îl afișează în pagina Web.

Partea de sfârșit a comentariului HTML este cea de lângă ultima linie:

```
//-->
```

Este precedată de două slashuri, ceea ce în scriptarea de partea clientului denotă o linie de comentariu. Cu alte cuvinte, eticheta finală de comentarii HTML este transformată în comentariu de browserele care suportă scriptare de partea clientului. Numai browserele care nu acceptă scriptarea de partea clientului folosesc această porțiune a etichetei de comentarii HTML.

Folosind eticheta de comentarii HTML de deschidere și de închidere, puteți ascunde cu succes codul aplicației dumneavoastră de browserele care nu acceptă scriptarea de partea clientului. Aceste tipuri de browsere ignoră complet și toate aparițiile etichetelor <SCRIPT> și </SCRIPT> deoarece toate browserele Web sunt hărăzite să ignore etichetele HTML pe care nu le recunosc.

Plasarea scriptului dumneavoastră în codul HTML

Există două modalități prin care puteți îngloba scripturile dumneavoastră VBScript într-o pagină Web. Prima modalitate este folosirea etichetelor <SCRIPT> și </SCRIPT> oriunde în documentul dumneavoastră HTML pentru a defini textul cuprins drept un script de partea clientului și nu ca text al documentului. A doua modalitate este de a îngloba codul VBScript direct într-o altă etichetă HTML. Desigur, această metodă funcționează numai cu etichetele în care codul de script înglobat este suportat.

Ați făcut deja cunoștință cu etichetele <SCRIPT> și </SCRIPT> în secțiunea „Eticheta <SCRIPT>", mai înainte în acest capitol. În acea secțiune, ați aflat că puteți plasa etichetele <SCRIPT> și </SCRIPT> oriunde în textul dumneavoastră HTML. Cel mai obișnuit loc de așezare a scriptului dumneavoastră este în secțiunea antet a HTML-ului, așa cum a fost definit prin etichetele <HEAD> și </HEAD>. Ați putea, apoi, plasa alte scripturi în secțiunea principală a documentului dumneavoastră HTML când doriți scrierea unui text specific la acea locație din pagina Web.

De exemplu, ați putea avea o subrutină în secțiunea de antet a documentului dumneavoastră HTML care să afișeze un șir în bara de stare oricând este apelată. Codul VBScript pentru funcție, completat cu etichetele HTML, este arătat în Listingul 24.1.

Listingul 24.1. Exemplu de subrutină VBScript în secțiunea HTML de antet.

```
<HEAD>
<TITLE>VBScript Example</TITLE>
<SCRIPT LANGUAGE="VBScript">
<!--
  Sub Status_Change(Status_String)
    status = Status_String
  End Sub
//-->
</SCRIPT>
</HEAD>
```

Subrutina primește un singur parametru șir, numit `Status_String`. Ea plasează apoi conținutul șirului în bara de stare a browserului Web.

A doua metodă de a îngloba cod VBScript în textul dumneavoastră HTML este plasarea codului în etichetele HTML suportate. Cele mai comune etichete HTML unde puteți plasa cod VBScript sunt eticheta <A>, eticheta <BODY> și eticheta <AREA>. Puteți avea, de asemenea, cod VBScript în etichetele HTML pentru creare de formulare.

Indiferent în care etichetă HTML alegeți să înglobați codul VBScript, sintaxa este totdeauna aceeași. Definiți atât limbajul de scriptare de partea clientului, cât și acțiunea de efectuat pentru un eveniment specificat. De exemplu, mai jos este o etichetă <BODY> care conține cod VBScript:

```
<BODY LANGUAGE="VBScript" onLoad="status='Finished'">
```

Ca și în cazul etichetei <SCRIPT>, este totdeauna nevoie să specificați opțiunea dumneavoastră de limbaj când folosiți VBScript. Altminteri, browserul Web folosește în mod prestabilit interpretorul JavaScript sau JScript.

A doua parte a muncii de adăugare a codului VBScript la o etichetă HTML este să folosiți subrutine de tratare a evenimentelor definite pentru acea etichetă și să le atribuiți cod VBScript. Subrutinele de tratare a evenimentelor sunt definite pentru anume etichete HTML, ca parte a modelului de obiect Internet Explorer (Internet Explorer Object Model), care este discutat în secțiunea „Modelul de obiect pentru scriptare Internet Explorer " mai departe, în acest capitol.

În acest caz, subrutina de tratare a evenimentului `onLoad` a fost folosită cu eticheta <BODY>. Evenimentul `Load` este asociat cu fereastra documentului. Când o nouă pagină Web este încărcată în browserul Web, este declanșat evenimentul `Load`. Folosind subrutina de tratare a evenimentului `onLoad`, puteți defini cod VBScript care intră în execuție de fiecare dată când are loc evenimentul `Load`.

Cu codul care urmează, comunicați browserului Web să plaseze șirul `Finished` în bara de stare a browserului Web când a terminat de încărcat pagina Web:

```
onLoad="status='Finished'"
```

De asemenea, ați putea plasa pur și simplu un apel la o subrutină sau funcție în subrutina de tratare a evenimentului, precum se arată în continuare:

```
onLoad="Status_Change('Finished')"
```

S-ar apela atunci subrutina `Status_Change` arătată mai devreme în Listingul 24.1.

Rularea scripturilor dumneavoastră de partea clientului

Programele dumneavoastră VBScript vor fi rulate automat de browserul Web care le încarcă, depinzând de modul cum le-ați integrat în pagina dumneavoastră Web. În mod normal, toate scripturile incluse cu eticheta HTML `<SCRIPT>` sunt analizate și executate când pagina Web este încărcată prima dată. Scripturile vor avea forma următoare:

```
<SCRIPT LANGUAGE="VBScript">
<!--
   status = "Thanks for visiting!"
//-->
</SCRIPT>
```

Acestă formă de scriptare este utilizată de obicei pentru inițializarea variabilelor și crearea datelor pe care le doriți înscrise direct în pagina Web. Folosind metoda obiectului `document.write`, puteți scrie date din scripturile dumneavoastră de partea clientului în pagina Web pentru a le afișa la utilizator. Însă puteți scrie numai în pagina Web în curs de încărcare în timp ce este analizată. Prin urmare, ar trebui să aveți toate instrucțiunile `document.write` în această formă:

```
<SCRIPT LANGUAGE="VBScript">
<!--
  document.write("You are using a Web browser that supports VBScript.")
//-->
</SCRIPT>
```

Instrucțiunile sunt executate în timp ce documentul este în curs de analiză, și datele sunt scrise în pagina Web.

Cealaltă opțiune pe care o aveți când scrieți cod pentru etichetele dumneavoastră `<SCRIPT>` și `</SCRIPT>` este să vă plasați codul în subrutine și funcții. Browserul Web nu rulează automat subrutinele și funcțiile când este analizată pagina Web. Ca urmare, trebuie să apelați subrutina sau funcția dumneavoastră într-un alt script pentru a fi rulată.

De exemplu, cu scriptul arătat în Listingul 24.1 ați definit subrutina `Status_Change`. Această subrutină nu este rulată automat când browserul Web încarcă pagina Web. De aceea, este necesar să apelați această subrutină undeva într-un alt script. De obicei, veți asocia cel de al doilea script cu un eveniment. Când evenimentul are loc, subrutina de tratare a evenimentului va apela subrutina dumneavoastră pentru a îndeplini acțiunea, precum în eticheta `<BODY>`:

```
<BODY LANGUAGE="VBScript" onLoad="Status_Change('Finished')">
```

ATENȚIE!

Subrutina de tratare a evenimentului `onLoad` este declanșată când pagina Web se încarcă complet. Dacă plasați, pur și simplu, un script care intră în execuție atunci când pagina Web este încărcată pentru a afișa mesajul de stare, el va fi transmis la bara de stare în timp ce pagina este încă analizată. Aceasta va face, în mod obișnuit, ca mesajul dumneavoastră să fie suprascris de browserul Web, care plasează mesajul `Done` în bara de stare când a terminat încărcarea paginii.

Modelul de obiect pentru scriptare Internet Explorer

În eforturile de a face scripturile de partea clientului mai puternice, Microsoft a definit un model de obiect pentru browserul său Internet Explorer. Acest model de obiect, cunoscut formal ca Modelul de obiect pentru scriptare Internet Explorer, definește o ierarhie de

obiecte, cu proprietățile, metodele și evenimentele asociate lor, pe care le puteți utiliza în codul dumneavoastră VBScript pentru a dezvolta aplicații mai robuste.

Modelul de obiect Internet Explorer pentru scriptare definește 10 obiecte:

- `Window` (fereastră)
- `Frame` (cadru)
- `History` (istoric)
- `Navigator` (navigator)
- `Location` (locație)
- `Document` (document)
- `Link` (legătură)
- `Anchor` (ancoră)
- `Form` (formular)
- `Element` (element)

Cele mai multe dintre aceste obiecte ar trebui să vă sune familiar. Ele reprezintă o versiune obiect a elementelor care compun un document pagină Web. De exemplu, aveți o fereastră de browser Web. Fiecare fereastră poate conține mai multe cadre. Fiecare pagină Web este referită în mod obișnuit ca un document, care conține legături, ancore, formulare și elemente de formular.

Unele dintre celelalte obiecte definesc elemente cu care, altminteri, este posibil să nu lucrați, exceptând scripturile de partea clientului. Fiecare fereastră de browser are, de asemenea, o listă cu istoric asociată ei, care face posibilă utilizarea butoanelor back (înapoi) și forward (înainte). Este posibil să doriți să controlați mișcarea în lista cu istoric, de exemplu, din scripturile dumneavoastră de partea clientului.

Obiectul `Location` reprezintă, pur și simplu, adresa URL a documentului încărcat curent în browserul Web. Aveți acces deplin la această informație. O puteți considera ca adresa întreagă sau să o spargeți în părțile asociate, cum ar fi protocolul, numele serverului sau adresa IP și numele documentului.

De asemenea, obiectul `Navigator` vă oferă informații despre browserul Web care este utilizat. El vă permite să descoperiți numele browserului, versiunea și platforma pe care rulează. Puteți folosi, ulterior, această informație pentru a prezenta un conținut personalizat, dependent de acești parametri.

Datorită Modelului de obiect, Microsoft a introdus o implementare alternativă pentru obiectul `Script`, care vă permite să folosiți etichetele `<SCRIPT>` și `</SCRIPT>` pentru a defini subrutine de tratare a evenimentelor pentru obiecte specifice. Această nouă implementare definește două atribute noi pentru eticheta `<SCRIPT>` și are următoarea formă:

```
<SCRIPT LANGUAGE=[language] FOR=[object] EVENT=[object's event]>
```

Pentru a utiliza această formă, plasați numele obiectului în atributul `FOR` și evenimentul obiectului pe care doriți să îl tratați în atributul `EVENT`. Internet Explorer execută conținutul acestui script doar când evenimentul acelui obiect ajunge să se declanșeze.

De exemplu, ați putea defini un câmp de introducere de text prin această linie:

```
<INPUT TYPE="text" NAME="txtName" SIZE=30>
```

Apoi, ați putea defini un script care să trateze evenimentul `Change`, care se declanșează ori de câte ori utilizatorul face o schimbare în conținutul câmpului text. Scriptul ar putea avea forma următoare:

```
<SCRIPT LANGUAGE="VBScript" FOR="txtName" EVENT="onChange">
```

```
<!--
  [Your script goes here]
//-->
</SCRIPT>
```

Sunteți deja familiarizat cu posibilitatea de a plasa subrutine de tratare a evenimentelor direct în eticheta HTML a obiectului care generează evenimentul. Ați putea trata acest eveniment similar prin folosirea sintaxei:

```
<INPUT TYPE="text" NAME="txtName" SIZE=30 onChange="[Your script goes here]">
```

Oricare ar fi modalitatea, acțiunea care are loc este aceeași. Evenimentul se declanșează, iar Internet Explorer caută o subrutină validă de tratare a evenimentului. Dacă este găsită una, se execută conținutul scriptului subrutinei de tratare a evenimentului.

Pentru ce este bun VBScript?

VBScript oferă autorilor de pagini Web un mod de a construi pagini bogate, mai interesante și mai funcționale decât ar fi putut să o facă doar cu HTML. HTML este o cale sublimă de a dezvolta documente ușoare pentru vizualizare pe tot felul de platforme. Totuși, el nu este un limbaj de programare, drept care produce pagini statice.

VBScript, pe de altă parte, este un limbaj de programare complet, cu instrucțiuni condiționale, structuri de date și structuri de control. Cu acestea puteți construi pagini Web care interacționează cu utilizatorul și execută o varietate de taskuri. Câteva taskuri comune efectuate cu JavaScript sunt: validarea datelor în formular, afișarea mesajelor în bara de stare a ferestrei browserului, configurarea și verificarea prăjiturilor, manipularea cadrelor și interacțiunea cu controale ActiveX. Aceste taskuri sunt descrise în secțiunile următoare.

Validarea datelor în formular

Una dintre cele mai comune aplicații ale VBScript este validarea introducerii de către utilizator a datelor de formular înainte ca introducerile să fie trimise serverului Web. Această formă de validare nu este altceva decât tipul de validare pe care ați fi îndeplinit-o, în mod normal, cu un script CGI, cum ar fi verificarea că în câmpurile obligatorii există date corespunzătoare și că datele care ar trebui să cadă în interiorul anumitor domenii sunt, într-adevăr, în interiorul respectivelor domenii. În principal, folosiți VBScript pentru a vă asigura că datele introduse de utilizator îndeplinesc un criteriu pe care l-ați stabilit pentru ele.

De exemplu, să presupunem că ați receptat reacții de la utilizatorii dumneavoastră și doriți să colectați numele și vârstele lor. Ați putea realiza un formular simplu pentru numele de familie al utilizatorului, prenume și vârstă. Ați putea crea, apoi, un program VBScript care verifică dacă utilizatorul a introdus date în toate trei câmpurile și că vârsta este într-o plajă rezonabilă, cum ar fi între 1 și 150 de ani.

Mai întâi, este nevoie să creați pagina Web care conține formularul HTML. Listingul 24.2 arată codul HTML pentru această pagină Web de exemplificare.

Listingul 24.2. Un exemplu de validare a formularului.

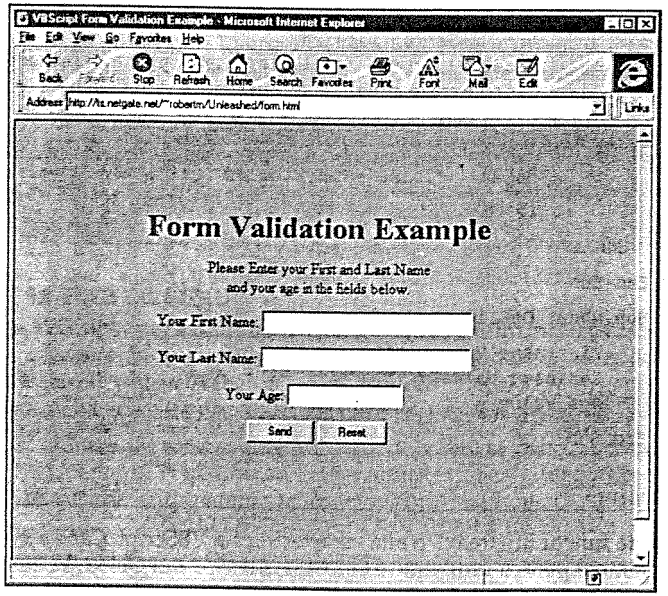
```
<HTML>
<HEAD>
<TITLE>VBScript Form Validation Example</TITLE>
</HEAD>
<BODY BGCOLOR="#00CC99" TEXT="#000000">
```



```
<TABLE BORDER=0 WIDTH=100% HEIGHT=100%>
<TR>
<TD VALIGN=center ALIGN=middle>
<H1>Form Validation Example</H1>
Please Enter your First and Last Name<BR>
and your age in the fields below.<BR>
<P>
Your First Name: <INPUT TYPE="text" NAME="Firstname" SIZE=40>
<P>
Your Last Name: <INPUT TYPE="text" NAME="Lastname" SIZE=40>
<P>
Your Age: <INPUT TYPE="text" NAME="Age" SIZE=20>
<P>
<INPUT TYPE="button" NAME="Send" VALUE="Send">
<INPUT TYPE="button" NAME="Reset" VALUE="Reset">
</TD>
</TR>
</TABLE>
</BODY>
</HTML>
```

Obesrvați lipsa etichetelor <FORM> și </FORM> din codul HTML prezentat în Listingul 24.2. Nu a fost nevoie să folosiți aceste etichete în prezentul exemplu deoarece veți folosi butoanele și câmpurile de introducere a textului intrinseci existente în Internet Explorer. Internet Explorer are elemente de formular incluse, de care puteți pace uz în loc de a vă construi formularul în modul tipic HTML. Figura 24.1 arată această pagină Web încărcată în Internet Explorer.

FIGURA 24.1. *Exemplu de validare a formularului în Internet Explorer.*



Întrucât nu folosiți eticheta <FORM> pentru a construi propriul formular HTML, nu aveți nici o locație pentru atributul ACTION. Totuși, aceasta nu este o problemă deoarece veți dezvolta o aplicație VBScript pentru a verifica datele formularului și a le returna către serverul Web după verificare.

Acum, că aveți pagina HTML care conține elementele formularului, este necesar să dezvoltați aplicația VBScript pentru a verifica datele. Nu este necesar ca scriptul dumneavoastră să efectueze vreo acțiune până când utilizatorul nu execută clic fie pe butonul Send, fie pe butonul Reset. Prin urmare, este nevoie să adăugați subrutine de tratare a evenimentelor, una pentru cazul în care se execută clic pe butonul Send și una pentru cel în care se execută clic pe butonul Reset. Codul pentru aceste subrutine de tratare a evenimentelor este în Listingul 24.3.

Listingul 24.3. Scriptul VBScript pentru exemplul de validare a formularului.

```
<SCRIPT LANGUAGE="VBScript">
<!--
    Sub Send_OnClick()
      If Firstname.Value = "" Or Lastname.Value = "" Or _
         Age.Value = "" Then
          MsgBox "You must enter values in all the fields", _
              16, "Empty Field!"
      Else
          If Age.Value >= 1 And Age.Value <= 150 Then
              MsgBox "Thank you for the information.", 0, "Thanks!"
              'Note: The following statement will send the data
              '      to the CGI script cgiscript.pl on the Web
              '      Server machine.
              location.href = "/cgi-bin/cgiscript.pl?Firstname=" & _
                  Firstname.Value & "&Lastname=" & Lastname.Value & _
                  "&Age=" & Age.Value
          Else
              MsgBox "Please enter a valid age!", 16, "Invalid Age!"
          End If
      End If
    End Sub
    Sub Reset_OnClick()
        Firstname.Value = ""
        Lastname.Value = ""
        Age.Value = ""
    End Sub
//-->
</SCRIPT>
```

Aceste funcții ilustrează o altă caracteristică a VBScript. Când ajunge să se declanșeze un eveniment pentru unul dintre obiectele din Modelul Obiect, VBScript caută o subrutină cu numele în forma următoare:

```
[object name]_On[object's event]()
```


Dacă VBScript găsește o subrutină care se potrivește acestei convenții de nume, el o execută ca subrutina de tratare a acelui eveniment.

Deci când utilizatorul execută clic pe butonul Reset, VBScript execută subrutina `Reset_OnClick`. Această subrutină este cât se poate de simplă. Ea reinițializează toate câmpurile, golindu-le.

Când utilizatorul execută clic pe butonul Send, se execută subprocedura `Send_OnClick`. Prima secțiune a acestei subrutine verifică dacă au fost introduse date în toate cele trei câmpuri. Dacă nu, este afișată o casetă de mesaj de eroare, care atrage atenția utilizatorului că trebuie introduse date în toate câmpurile. Această casetă de mesaj este arătată în Figura 24.2.

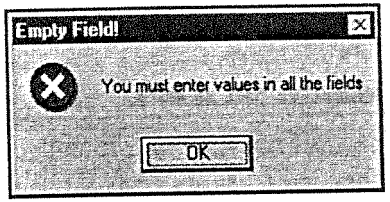


FIGURA 24.2. *Caseta de mesaj Empty Field!.*

Când utilizatorul a introdus date în toate câmpurile, se execută partea `Else` a instrucțiunii `If...Then...Else`. Prima instrucțiune din acest bloc este o altă instrucțiune `If... Then...Else`, care verifică domeniul valorii introduse de utilizator în câmpul `Age` (Vârsta). Dacă datele nu sunt între numerele 1 și 150, se afișează un mesaj de eroare atrăgând atenția utilizatorului să introducă o vârstă validă.

În final, când utilizatorul a introdus date valide pentru toate câmpurile, se afișează caseta de mesaj de mulțumire și datele sunt trimise unui script CGI de pe serverul Web cu următoarea instrucțiune:

```
location.href = "/cgi-bin/cgiscript.pl?Firstname=" & _
  Firstname.Value & "&Lastname=" & Lastname.Value & _
  "&Age=" & Age.Value
```

De fapt, această instrucțiune lucrează ca o redirectare, reclamând ca browserul Web să încarce adresa URL cerută. În acest caz, adresa URL cerută este un script CGI, iar datele din formularul utilizatorului sunt adăugate la sfârșitul adresei URL ca un șir de interogare.

ATENȚIE

Dacă folosiți o procedură similară pentru a valida datele din formularul utilizatorului și a le trimite scriptului dumneavoastră CGI, țineți cont că scriptul dumneavoastră CGI va primi datele cu metoda de interpelare GET, nu POST.

Afișarea mesajelor în bara de stare

O altă utilizare comună a VBScript este să ofere o reacție inversă către utilizatorii dumneavoastră în bara de stare a ferestrei browserului. Aceste mesaje sunt, adesea, doar scurte prezentări de informații legate de pagina Web, legături pe pagină ori situl Web în general. O aplicație răspândită este să folosiți bara de stare ca pe un mesaj cu text defilant.

În câteva din exemplele anterioare din acest capitol, ați văzut deja cum să plasați un mesaj în bara de stare din Internet Explorer. Totuși, nu au existat multe explicații despre ceea ce era într-adevăr de făcut. În exemplele precedente, ați observat linii similare acesteia:

```
status = "Status Bar Message"
```

Acesta este, de fapt, doar un mod prescurtat de a stabili mesajul din bara de stare a ferestrei. Sintaxa formală arată astfel:

```
windows.status = "Status Bar Message"
```

Această sintaxă ilustrează mai bine folosirea Modelului de obiect Internet Explorer. Așa cum ați învățat în secțiunea „Modelul de obiect pentru scriptare Internet Explorer", `Window` este un obiect din ierarhia obiectelor. Una dintre proprietățile asociate cu obiectul `Window` este `status`, care stochează cuprinsul mesajului din bara de stare. Atribuind o valoare nouă proprietății, modificați mesajul care este afișat.

Versiunea prescurtată utilizată anterior este un alt mod prin care puteți stabili mesajele din bara de stare. Când este folosită versiunea prescurtată, Internet Explorer utilizează fereastra curentă ca opțiune prestabilită. Versiunea prescurtată este utilizată pentru mesajul cu text defilant din bara de stare care este prezentat în listingul 24.4. Figura 24.3 arată cum Internet Explorer afișează această pagină Web.

Listingul 24.4. Mesajul cu text defilant din bara de stare.

```
<HTML>
<HEAD>
<TITLE>A Status Bar Scrolling Marquee</TITLE>
<SCRIPT Language="VBScript">
<!--
  Dim preceding_spaces
  Dim scrolling_message
  Dim current_message

  preceding_spaces = Space(130)
  scrolling_message = "This message will move from " & _
    "the right to the left side of the status bar..."
  current_message = ""

  Sub Scroll_Message()
    If Len(preceding_spaces) > 1 Then

      preceding_spaces = Right(preceding_spaces, Len(preceding_spaces) - 1)
      If Len(current_message) < Len(scrolling_message) Then
        current_message = current_message & _
          Mid(scrolling_message, Len(current_message) + 1, 1)
      End If

    Else

      If Len(current_message) = 1 Then

        preceding_spaces = Space(130)
        current_message = ""

      Else

        current_message = Right(current_message, _
          Len(current_message) -1)

      End If

    End If

    status = preceding_spaces & current_message
    MyID = setTimeout("Scroll_Message", 150, "VBScript")

  End Sub

//-->
</SCRIPT>
</HEAD>
```



```
<BODY BGCOLOR="#FFCC00" LANGUAGE="VBScript" onLoad=" Scroll_Message">
<TABLE BORDER=0 WIDTH=100% HEIGHT=100%>
<TR>
<TD VALIGN=center ALIGN=middle>
<FONT FACE="COURIER" SIZE=+1>This page contains an example<BR>
of placing a scrolling marquee<BR>
message in Internet Explorer's<BR>
status bar.</FONT>
</TD>
</TR>
</TABLE>
</BODY>
</HTML>
```

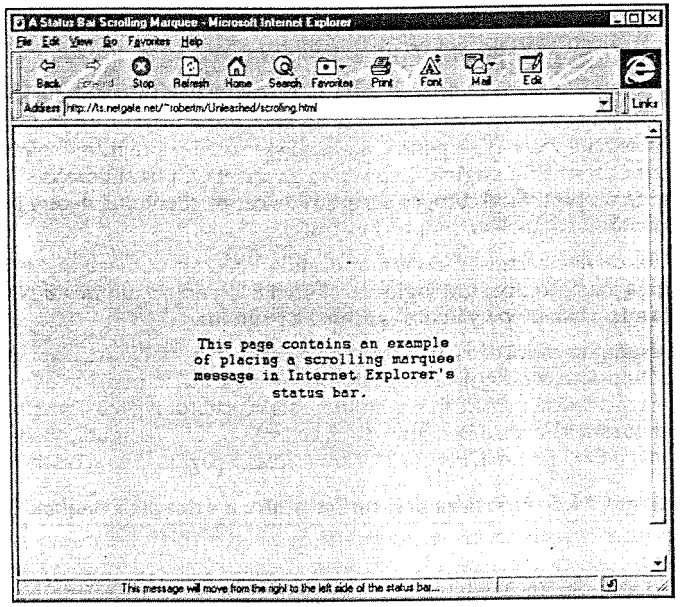


FIGURA 24.3. *Exemplu de mesaj cu text defilant în bara de stare în Internet Explorer.*

Porțiunea de script a Listingului 24.4 începe prin a defini trei variabile cu instrucțiunea `Dim`:

```
Dim preceding_spaces
Dim scrolling_message
Dim current_message
```

După declararea lor, fiecăreia i se dă o valoare inițială. Variabila `preceding_spaces` conține numărul spațiilor de plasat înainte de mesaj. Pentru a face ca mesajul să defileze de-a lungul barei de stare, scriptul dumneavoastră pornește cu o mulțime de spații premergătoare, iar apoi reduce în mod repetat numărul de spații cu unu, până când mai rămâne numai unul.

Pentru a părea că mesajul intră din partea dreaptă a barei de stare și dispare la stânga, este nevoie și să adăugați doar o porțiune din mesaj la un anume moment până ce mesajul apare complet în bara de stare. După aceea, când toate spațiile premergătoare au dispărut, începeți să ștergeți caractere de la începutul șirului, până când acesta se golește. Pentru a face aceasta, se folosește variabila `current_message` pentru a stoca porțiunea de mesaj

defilant care se află în mod curent în bara de stare. Întregul mesaj de defilat este stocat în variabila `scrolling_message`.

Subrutina `Scroll_Message` este utilizată pentru a determina noua porțiune a mesajului care va fi afișată în bara de stare. Pentru ca mesajul să defileze, trebuie să fie afișată repetat o combinație de spații premergătoare și caractere din mesajul defilant. Depinde de subrutina `Scroll_Message` să determine combinația și să afișeze mesajul. După afișarea unei iterații de spații și caractere din mesajul defilant, ea folosește linia următoare:

```
MyID = setTimeout("Scroll_Message", 150, "VBScript")
```

Această instrucțiune face ca programul să ia o pauză de 150 milisecunde. După pauză, subrutina `Scroll_Message` este executată din nou. Dacă nu aveți pauză între iterații în programul dumneavoastră VBScript, mesajul va defila prea repede, iar utilizatorii dumneavoastră nu vor fi capabili să îl citească. Dacă doriți să creșteți sau să scădeți viteza de defilare, reduceți sau creșteți numărul 150 din această instrucțiune.

Lucrul cu prăjituri

Capacitatea de a stabili și a verifica prăjituri direct pe mașina clientului este un avantaj important al scripturilor de partea clientului. Mai înainte, trebuia să folosiți un script CGI pe mașina server Web pentru a stabili și vizualiza prăjituri. Înainte trebuia să formulați cererile simple de pagină Web printr-un script CGI dacă doreați să utilizați o prăjitură cu pagina respectivă. Acum, cu scripturi de partea clientului, fiecare pagină Web este împuternicită să controleze prăjituri ea însăși.

La fel ca JavaScript și JScript, aplicațiile VBScript pot stabili și verifica prăjituri pentru pagina Web în care sunt înglobate. Puteți folosi aceste informații pentru a afișa, apoi, date dinamice, bazate pe valoarea curentă a prăjiturii.

De exemplu, codul din Listingul 24.5 produce o pagină Web cu o aplicație VBScript înglobată, care folosește o prăjitură. Acest exemplu verifică dacă prăjitura a fost stabilită. Dacă nu, se afișează o casetă de mesaj care mulțumește vizitatorului pentru vizitarea sitului. Prăjitura este configurată să expire la o lună după ce este stabilită, încât utilizatorul vede caseta de mesaj numai o dată pe lună. Figura 24.4 arată această pagină Web și caseta de mesaj.

Listingul 24.5. Folosirea prăjiturilor pentru a determina conținutul.

```
<HTML>
<HEAD>
<TITLE>Using Cookies to Determine Web page Content</TITLE>
<SCRIPT LANGUAGE="VBScript">
<!--
  cookie_index = Instr(Document.Cookie, Visited)
  If cookie_index = 0 Then
    Document.Cookie = Visited & "=" &"1;expires=" & DateAdd("m", 1, Now)
    MsgBox "Thank you for visiting our site!"
  End If
//-->
</SCRIPT>
</HEAD>
<BODY BGCOLOR="#00CCFF">
<TABLE BORDER=0 WIDTH=100% HEIGHT=100%>
<TR>
<TD VALIGN=center ALIGN=middle>
<FONT FACE="COURIER" SIZE=+1><B>Have you been here before?</B><BR>
```



```
This page displays a message box<BR>
if you have not previously<BR>
visited this site.</FONT>
</TD>
</TR>
</TABLE>
</BODY>
</HTML>
```

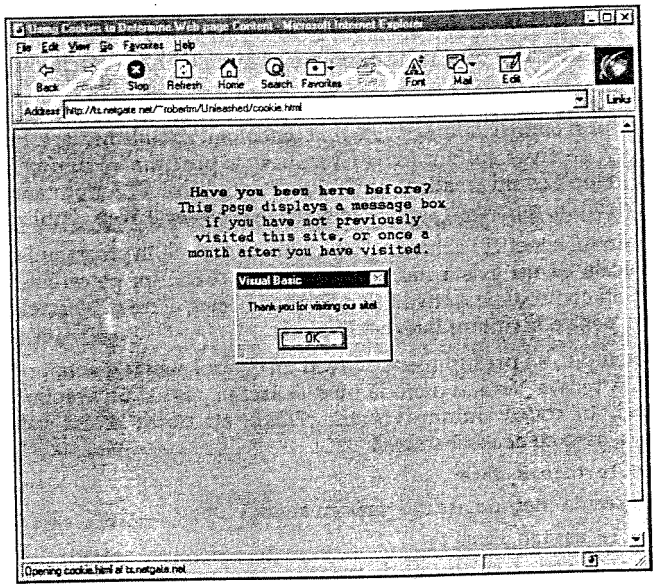


FIGURA 24.4. *Exemplul cu prăjitură în Internet Explorer.*

Lucrul cu controale ActiveX

Veți remarca adevărata putere a lui VBScript când îl veți folosi cu controale ActiveX. VBScript, deși capabil să realizeze unele aplicații de bază prin el însuși, are unele limitări majore. Cele mai multe dintre aceste limitări există datorită incapacității sale de a accesa sistemul de operare sau sistemul de fișiere. În mod natural, aceste limitări sunt necesare, întrucât deschiderea acestor sisteme de către VBScript ar prezenta un mare risc de securitate.

ActiveX, pe de altă parte, nu posedă aceste limitări. Controalele ActiveX pot face schimbări în sistemul de fișiere și pot accesa sistemul de operare pentru a realiza taskurile sale. Prin folosirea de certificate de autentificare cu controale ActiveX, Microsoft pretinde că utilizatorii se pot simți încrezători în ceea ce privește securitatea controalelor ActiveX publicate de surse reputate.

Ce sunt controalele ActiveX?

ActiveX este, de fapt, numele unui set de tehnologii create de Microsoft. Aceste tehnologii sunt destinate pentru a fi multiplatformă, extensibile și optimizate pentru Internet. Respectiva optimizare constă în dimensiune mică, descărcare automată și certificate de autentificare pentru scopuri de securitate.

De fapt, ActiveX refolosește o tehnologie existentă. Este următoarea întruchipare a tehnologiei Microsoft OLE, care a fost prima dată introdusă în 1991. OLE a permis utilizatorilor

să înglobeze obiecte de tipuri diferite într-un singur fișier. Folosind OLE, puteați îngloba foi de calcul tabelar Excel în documente Word sau, invers, să înglobați documente Word în foi de calcul tabelar Excel, spre exemplu.

Tehnologia fundamentală atât a ActiveX, cât și a OLE, o constituie Modelul de obiect component Microsoft (COM, Component Object Model). COM este un standard prin care obiectele comunică unul cu altul. Bazându-se pe COM, producătorii pot crea obiecte care comunică unul cu altul fără a trebui să aibă grijă de detaliile despre cum funcționează comunicarea.

Deși Microsoft a intenționat ca ActiveX să fie multiplatformă, ea este disponibilă curent doar pe o singură platformă, Windows. Microsoft lucrează cu alte companii să porteze ActiveX către alte platforme populare, precum UNIX și Macintosh.

VBScript și controalele ActiveX

VBScript și controalele ActiveX sunt tehnologii complementare. Puteți dezvolta controale ActiveX de nivel inferior folosind limbaje de programare cum ar fi Java, C, C++ și Visual Basic. După ce ați finalizat controlul, puteți folosi VBScript pentru a accesa și modifica proprietățile, metodele și evenimentele în browserul Web al utilizatorului.

Adevăratul beneficiu al lucrului cu controale ActiveX în programul dumneavoastră VBScript este acela că nu există nici o tehnică nouă VBScript pe care este nevoie să o învățați. Accesați controalele ActiveX în același fel în care accesați obiectele definite în Modelul de obiect pentru scriptare Internet Explorer.

De exemplu, să presupunem că aveți un control ActiveX numit `schedule` cu o proprietate numită `todays_date`, o metodă numită `assign_task` și un eveniment numit `meeting_time`. Atunci, în codul dumneavoastră HTML, ați putea accesa valoarea din proprietatea `todays_date` cu această sintaxă:

```
schedule.todays_date
```

Veți executa metoda `assign_task` cu această linie:

```
schedule.assign_task ()
```

Și, dacă ați scris o subrutină de tratare a evenimentului pentru evenimentul `meeting_time`, numele subrutinei va fi următorul:

```
schedule.Onmeeting_time
```

Acesta este exact același mod în care accesați proprietăți, metode și evenimente pentru celelalte obiecte definite în Modelul de obiect Internet Explorer. Natural, proprietățile, metodele și evenimentele concrete variază, depinzând de controlul ActiveX. Spre a scripta pentru ele, trebuie să cunoașteți numele lor pentru controlul ActiveX pe care doriți să îl utilizați.

Resurse pentru o informare sporită

Acest capitol a oferit doar un mic eșantion de limbaj de scriptare VBScript și de ceea ce puteți realiza cu el. Această secțiune vă arată cum să obțineți mai multe informații despre folosirea VBScript în paginile dumneavoastră Web. Este împărțită în două subsecțiuni: una despre resurse VBScript online, iar cealaltă despre resurse offline.

Resurse online

Toate informațiile de care aveți nevoie pentru a începe dezvoltarea aplicațiilor VBScript destinate paginilor dumneavoastră Web se regăsesc online într-o formă sau alta. Cele mai multe dintre informații pot fi găsite în paginile Web despre VBScript de la Microsoft. Totuși, multe alte resurse Internet oferă informații folositoare despre VBScript și tehnologie asociată.

Situri Web

De departe, cea mai bogată sursă de informații online despre VBScript și tehnologiile asociate sunt siturile Web. Cele mai multe dintre situri oferă îndrumare (tutorials), referințe lingvistice și o grămadă de scripturi de exemplificare. Ceea ce urmează este o listă de pagini Web folositoare pentru VBScript, precum și pentru tehnologiile HTML, JScript, ActiveX și de prăjitură asociate:

HTML Quick Reference

`http://www.cc.ukans.edu/info/HTML_quick.html`

HyperText Markup Language

`http://www.w3.org/pub/WWW/MarkUp/`

A Beginner's Guide to HTML

`http://www.ncsa.uiuc.edu/General/Internet/WWW/HTMLPrimer.html`

Special Characters

`http://www.utoronto.ca/webdocs/HTMLdocs/NewHTML/entities.html`

ISO8859-1 (Latin-1) Table

`http://www.uni-passau.de/-ramsch/iso8859-1.html`

Microsoft's VBScript Web Pages

`http://www.microsoft.com/vbscript/`

Scribe – The VBScript Writers Resource

`http://www.km-cd.com/scribe/`

NCompass Labs

`http://www.ncompasslabs.com/products.htm`

Ask the VB Pro

`http://www.inquiry.com/techtips/thevbpro/`

Rollins & Associates Home Page

`http://www.rollins-assoc.com/`

Microsoft's JScript Web Pages

`http://www.microsoft.com/jscript/`

Compatibility Issues of Using JavaScript with Microsoft Internet Explorer 3.0

`http://www.gmccomb.com/javascript/ie30.html`

Microsoft's ActiveX Pages

`http://www.microsoft.com/activeplatform/default.asp`

ZDNet's ActiveX Files

`http://www5.zdnet.com/zdwebcat/content/activex/`

ClNet's ActiveX Resources Pages

`http://www.activex.com/`

Netscape's Persistent Cookies Documentation

`http://home.netscape.com/newsref/std/cookie_spec.html`

Malcolm's Guide to Persistent Cookies

`http://www.emf.net/-mal/cookiesinfo.html`

Andy's Netscape HTTP Cookie Notes

`http://www.illuminatus.com/cookie`

Noi pagini Web apar tot timpul, deci o listă dintr-o carte publicată nu poate constitui un set complet de referințe pentru siturile Web. De aceea, dacă sunteți în căutare de informații legate de VBScript negăsite în precedentele surse, ar trebui să faceți uz de unul dintre cataloagele de căutare din Internet, cum ar fi următoarele:

Yahoo!(`http://www.yahoo.com/`)

Infoseek(`http://www.infoseek.com/`)

AltaVista(`http://www.altavista.digital.com/`)

Excite(`http://www.excite.com/`)

Lycos(`http://www.lycos.com/`)

ClNet's Search.com(`http://www.search.com/`)

Hotbot(`http://www.hotbot.com/`)

Grupuri de discuții și liste poștale

Suplimentar bogăției de resurse legate de Web, Internet posedă și zone unde pot fi transmise întrebări cărora să li se răspundă de către alți utilizatori bine informați. Aceste zone sunt grupuri de discuții și liste poștale prin listserv. Ambele sunt resurse excelente de aflare a răspunsurilor exacte la întrebări pe care le aveți despre VBScript. Ceea ce urmează este o listă a grupurilor de discuții legate de subiect și a listelor poștale listserv pe care le puteți considera utile:

`comp.infosystems.www.authoring.html`

`comp.infosystems.www.browsers.mac`

`comp.infosystems.www.browsers.ms-windows`

`comp.infosystems.www.browsers.x`

`comp.infosystems.www.browsers.misc`

`http://www.dejanews.com/` (Deja News este un serviciu care caută grupuri de discuții Usenet după cuvinte cheie.)

Lista poștală VBScript

Pentru a subscrie, trimiteți un mesaj prin e-mail la: `Listserv@listserv.msn.com`.

În corpul mesajului, tastați `Subscribe VBScript Firstname Lastname`.

Lista poștală Controale ActiveX

Pentru a subscrie, trimiteți un mesaj prin e-mail la: `Listserv@listserv.msn.com`.

În corpul mesajului, tastați `Subscribe ActiveXControls Firstname Lastname`.

Lista de discuții HTML avansate

Pentru a subscrie, trimiteți un mesaj prin e-mail la: `Listserv@ua1vm.ua.edu`.

În corpul mesajului, tastați `Subscribe ADV-HTML Firstname Lastname`.

Resurse offline

Cele mai multe dintre resursele VBScript disponibile curent sunt resurse online, cum ar fi situri Web, grupuri de discuții și liste poștale listserv. Ele sunt cele mai bune surse ale dumneavoastră pentru informare curentă și exemple care tratează VBScript.

Pentru aceia care preferă sursele de informare offline, cea mai bună resursă ar fi variatele cărți de calculatoare scrise despre VBScript. Printre cele mai notabile titluri se numără:

Special Edition, Using VBScript, QUE

Laura Lemay's Web Workshop ActiveX and VBScript, Sams.net Publishing

How to Program Microsoft Visual Basic Scripting Edition, Ziff-Davis Press

The Comprehensive Guide to VBScript: The Encyclopedic Reference for VBScript, HTML, & ActiveX, Ventana Communications Group Inc.

VBScript Unleashed, Sams.net Publishing

Teach Yourself VBScript in 21 Days, Sams.net Publishing

Dacă sunteți interesat de achiziționarea unei cărți despre VBScript, întrebați de unul dintre aceste titluri sau de alte titluri la raionul cu cărți de calculatoare de la cea mai apropriată librărie. Puteți, de asemenea, să căutați și să cumpărați cărți VBScript de la cele mai populare magazine de cărți online, cum ar fi Amazon (`http://www. amazon.com`) și Barnes and Noble (`http://www.barnesandnoble.com`)

Rezumat

Visual Basic Scripting Edition, sau VBScript, este un limbaj de scriptare de partea clientului care folosește mult din aceeași sintaxă și structură ca și popularul limbaj de programare Microsoft Visual Basic. Microsoft a introdus VBScript pentru a oferi programatorilor experimentați în Visual Basic un mod ușor de a scrie aplicații pentru pagini Web.

VBScript nu este doar o versiune de scriptare a lui Visual Basic. Datorită elementelor de securitate, unele caracteristici Visual Basic nu sunt acceptate în VBScript. În plus, datorită mediului unic în care rulează aplicațiile VBScript, există unele caracteristici noi, specifice lui VBScript, care nu aparțin produsului Visual Basic. Totuși, ori de câte ori a fost posibil, caracteristicile Visual Basic au fost acceptate în VBScript pentru a face cele două limbaje cât de asemănătoare posibil.

Ca și alte limbaje de scriptare de partea clientului, VBScript a intenționat să fie înglobat în codul sursă HTML al paginilor Web. El acceptă aceleași etichete `<SCRIPT>` și `</SCRIPT>` pentru înglobarea codului în documentul HTML. Codul VBScript poate, de asemenea, să fie înglobat în mai multe etichete HTML care reprezintă obiecte din Modelul de obiecte pentru scriptare Internet Explorer.

Există o largă varietate de aplicații pentru VBScript. Unele dintre cele mai comune aplicații sunt pentru validarea formularului, mesaje în bara de stare și manipularea cadrelor. Deși folositor, adevărata putere a lui VBScript apare în capacitatea de a lucra cu controale ActiveX. Tot așa cum poate manipula obiecte în Modelul de obiecte, VBScript permite scriptarea pentru controale ActiveX, astfel încât autorii de pagini Web să controleze dinamic controalele lor ActiveX înglobate.

Capitolul 25
Să înțelegem interfața CGI

de Robert McDaniel

În acest capitol

- **Ce este CGI? 352**
- **Integrarea aplicațiilor CGI în situl dumneavoastră Web 354**
- **Pentru ce este bună CGI? 359**
- **Când se utilizează CGI? 360**
- **Resurse pentru o informare sporită 375**

La început World Wide Web a constat din documente hipertext statice, care, în forma lor originală, nici măcar nu suportau imagini inline. Browserele Web afișau pagina de text cu formatarea limitată oferită de primele etichete HTML. În acele zile, nimeni nu realiza impactul pe care Web avea să îl producă în curând asupra lumii. Totuși, chiar și atunci Web s-a bucurat de multă atenție.

Pe măsură ce mai multă lume devenea interesată de această nouă tehnologie, apăreau tot mai multe sugestii pentru adăugarea de caracteristici. Multe dintre noile caracteristici au fost integrate de partea clientului în browserul Netscape Navigator. Printre aceste caracteristici s-au numărat adăugarea de imagini GIF inline, mai multe etichete de machetare și tabelele. O mulțime de caracteristici totuși nu au putut fi implementate de partea clientului. Atunci s-a descoperit că este necesară capacitatea de a integra aplicații în serverul Web. Permițând dezvoltatorilor să procedeze astfel, programatorii care codau software-ul de server Web nu ar fi trebuit să facă toate modificările pe care le doreau dezvoltatorii; acestea puteau fi adăugate de dezvoltatori prin aplicații personalizate.

Common Gateway Interface (interfața de poartă comună), sau CGI, a fost primul mod prin care puteați integra aplicații în situl dumneavoastră Web. Prin intermediul CGI, dezvoltatorii puteau scrie propriile lor aplicații, personalizate pentru taskul pe care doreau să îl efectueze. Puteau ulterior să integreze ușor aceste aplicații în situl lor Web, indiferent de software-ul care rula pe serverul Web.

Cu toate că CGI există de ceva vreme, reprezintă, încă, un mijloc viabil de dezvoltare a aplicațiilor pentru utilizare cu un sit Web. În acest capitol, faceți cunoștință cu CGI prin intermediul următoarelor subiecte:

- Ce este CGI
- Când să utilizați CGI
- Cum să integrați aplicațiile CGI și paginile dumneavoastră HTML
- Pentru ce este bună CGI
- Resurse pentru o informare sporită

Ce este CGI?

Mai întâi, trebuie să știți că CGI nu este un limbaj de programare. Este o poartă prin care comunică servere Web și alte aplicații. CGI definește un mod standard în care datele pot fi trecute de la un server Web către o aplicație CGI și apoi înapoi la serverul Web, după ce aplicația CGI s-a terminat.

Aplicațiile CGI rulează ca aplicații separate de serverul Web, așa încât nu au acces la toate informațiile la care serverul Web are, precum versiunea de browser Web și numele sau oricare date din formulare pe care utilizatorul le trimite serverului Web. Totuși, aplicația dumneavoastră trebuie să aibă acces la aceste informații pentru a-și îndeplini lucrarea.

De asemenea, după ce aplicația CGI și-a terminat execuția, ea are, de obicei, nevoie să returneze rezultatele taskului către browserul Web al utilizatorului. Prin urmare, trebuie să existe o cale pentru ca datele să meargă de la aplicația CGI la browserul Web al utilizatorului. Cel puțin, aplicația CGI trebuie să fie capabilă să trimită datele înapoi către serverul Web, care le îndreaptă apoi către browserul Web al utilizatorului.

Fluxul datelor dintre serverul Web și aplicația dumneavoastră CGI este miezul programării CGI. O dată ce știți cum să faceți aceasta, sunteți gata să începeți dezvoltarea propriilor dumneavoastră aplicații CGI. Acest subiect este discutat în secțiunea „Integrarea aplicațiilor CGI în situl dumneavoastră Web", mai târziu, în acest capitol. Înainte de a începe dezvoltarea aplicațiilor CGI, aveți neapărată nevoie să alegeți un limbaj de programare sau de scriptare pentru a scrie aplicații CGI.

Alegerea unui limbaj de programare

Întrucât CGI nu este un limbaj de programare, trebuie să alegeți unul pentru codificarea aplicațiilor dumneavoastră CGI. Alegerea unui limbaj de programare pentru crearea aplicației executabile care poate fi rulată pe mașina serverului dumneavoastră Web este condiționată doar de disponibilitatea limbajului respectiv. Din punct de vedere practic, totuși există unele considerații de luat în seamă la alegerea limbajului aplicației dumneavoastră CGI.

În primul rând este bine să alegeți un limbaj folosit în mod obișnuit și de alți producători de aplicații CGI. Astfel de limbaje sunt Perl, C, C++, programe shell UNIX, Java, TCL, Visual Basic și AppleScript. Prin alegerea unuia dintre aceste limbaje, puteți beneficia de avantajul suportului comunității producătorilor din Web atunci când dezvoltați aplicațiile dumneavoastră proprii.

Există și multe arhive care conțin aplicații CGI și biblioteci gata făcute, pe care le puteți folosi gratuit în situl propriu. Aceste arhive conțin, de obicei, scripturi în limbajele menționate. Folosirea acestui cod existent ori de câte ori este posibil vă va reduce timpul de dezvoltare.

În al doilea rând, este bine să alegeți un limbaj de programare adecvat pentru taskul de efectuat. Dacă executați acțiuni care utilizează intensiv procesorul, precum căutarea și sortarea în milioane de înregistrări, ar trebui să folosiți un limbaj de programare compilat, precum C sau C++, care vă oferă randament mult mai bun decât un limbaj interpretat, cum ar fi Perl. Totuși, dacă randamentul nu este o problemă pentru aplicația pe care o codificați, atunci un limbaj de scriptare precum Perl vă poate reduce semnificativ timpul de dezvoltare, deoarece cei mai mulți programatori pot scrie un cod similar în Perl mult mai iute decât în C.

Aveți, de asemenea, în vedere platforma pe care intenționați să ruleze aplicația dumneavoastră. Evident, dacă aveți de gând să rulați aplicația pe o mașină UNIX, atunci AppleScript nu este o opțiune pentru dumneavoastră. Totuși, s-ar putea să nu doriți să utilizați un limbaj acceptat pe o platformă dată din cauza problemelor specifice acelei platforme. De exemplu, Java rulează rezonabil pe mașini UNIX. Pe mașini Macintosh totuși el rulează penibil de încet. Prin urmare, dacă platforma țintă este Macintosh, Java nu ar reprezenta o alegere bună.

Aceleași considerente se păstrează pentru sistemele de operare. Unele sisteme de operare nu rulează limbajele interpretate foarte bine. De exemplu, Perl reprezintă o alegere populară pentru scrierea aplicațiilor CGI. Totuși, când rulează pe o mașină cu sistem de operare Windows NT, randamentul poate fi foarte scăzut. În consecință, veți dori să alegeți un limbaj diferit.

Integrarea aplicațiilor CGI în situl dumneavoastră Web

După ce începeți dezvoltarea aplicațiilor CGI, veți realiza curând că pentru a integra o aplicație în situl dumneavoastră Web, trebuie să se întâmple două lucruri. Mai întâi, aplicația dumneavoastră trebuie să fie capabilă de a recepționa date de la browserul Web și de la serverul Web. Acest lucru este imperativ pentru a îndeplini corect cele mai multe funcții. În al doilea rând, aplicația dumneavoastră trebuie să fie capabilă să trimită rezultate înapoi browserului Web al utilizatorului după ce și-a încheiat execuția.

CGI definește cum se realizează aceste două taskuri. Pentru a recepționa date de la browserul și serverul Web, aplicația dumneavoastră CGI nu are de făcut nimic pentru a se asigura că aceste date sunt disponibile. Serverul Web trimite automat aceste informații către aplicația dumneavoastră CGI când este pornită. Și trimiterea datelor înapoi către browserul Web este la fel de ușoară. Tot ce aveți de făcut este să creați un antet valid, așa cum este definit în specificația CGI, și să transmiteți rezultatele.

Conexiuni HTTP și anteturi

În miezul tuturor interacțiunilor Web se află conexiunea prin protocolul de transfer al hipertextelor, Hypertext Transfer Protocol, sau HTTP. Prin conexiunea HTTP, browserele Web cer documente pe care serverele Web le returnează. Când un browser Web cere un document, el trimite o cerere HTTP către serverul Web pe care se află documentul. Când serverul Web recepționează o cerere HTTP, el returnează documentul într-un răspuns HTTP, dacă acesta există și utilizatorului îi este permis să aibă acces la el. Cererea și răspunsul HTTP alcătuiesc o conexiune HTTP.

Toate cererile și răspunsurile HTTP conțin o secțiune cunoscută ca *antet HTTP* (HTTP header). Antetul HTTP include informații despre document, serverul Web, browserul Web și alte elemente asociate. Aceste informații sunt stocate în câmpuri individuale în interiorul antetului. Anteturile HTTP nu sunt la fel pentru cerere și răspuns; totuși, unele dintre câmpuri pot fi la fel.

Câmpurile antetului de cerere HTTP sunt arătate în Tabelul 25.1. Ele sunt câmpurile definite pentru ca un browser Web să le trimită într-un antet de cerere HTTP.

Tabelul 25.1. Anteturi de cerere HTTP.

Câmpul antetului de cerere	*Semnificație*
ACCEPT	Câmpul ACCEPT conține o listă de tipuri de suport, în format MIME, pe care browserul Web le poate accepta de la serverul Web.
ACCEPT-ENCODING	Câmpul ACCEPT-ENCODING este utilizat pentru a specifica serverului Web metodele de codificare a documentului pe care browserul Web le acceptă.
ACCEPT-LANGUAGE	Specifică serverului Web limba preferată de browserul Web pentru răspuns.

Câmpul antetului de cerere	*Semnificație*
AUTHORIZATION	Stochează informația de autentificare trimisă de browserul Web spre serverul Web pentru a se autoidentifica.
CHARGE-TO	Câmpul CHARGE-TO nu este implementat în mod obișnuit.
FROM	Indică adresa e-mail a utilizatorului al cărui browser Web a trimis cererea HTTP. Cele mai multe browsere curente nu furnizează o valoare pentru acest câmp.
IF-MODIFIED-SINCE	Este o valoare de tip dată care comunică serverului Web să returneze documentul numai dacă s-a modificat de la respectiva dată.
PRAGMA	Câmpul PRAGMA conține instrucțiuni speciale pentru serverul Web. De exemplu, directiva PRAGMA no-cache instruiește serverele Web să nu trimită versiuni stocate în cache ale documentului cerut.
REFERER	Furnizează adresa URI a documentului în care utilizatorul a executat clic pe o legătură pentru a naviga spre documentul curent.
USER-AGENT	Stochează numele și versiunea de browser Web care produce cererea. De exemplu, `Mozzila/4.01[en] (Win95; I)` este agentul de utilizator pentru Netscape 4.0 pe platformă Windows 95.

În mod natural, cele mai multe informații din antetul cererii se referă la browserul Web. Deoarece browserul Web face cererea, el are nevoie să transmită informații relevante serverului Web. Când serverul Web răspunde, el formează un antet de răspuns HTTP, alcătuit din câmpurile listate în Tabelul 25.2.

Tabelul 25.2. Anteturi de răspuns HTTP.

Câmpul antetului de răspuns	*Semnificație*
ALLOWED	Comunică browserului Web ce metode de cerere sunt permise, cum ar fi metoda GET.
CONTENT-ENCODING	Specifică browserului Web metoda de codificare a documentului folosită pentru documentul returnat.
CONTENT-LANGUAGE	Specifică limba din documentul returnat.
CONTENT-LENGTH	Conține dimensiunea, în octeți, a documentului care este returnat.
CONTENT-TRANSFER-ENCODING	Conține codificarea datelor dintre serverul Web și browserul Web, cum ar fi în binar sau ASCII.
CONTENT-TYPE	Specifică browserului Web tipul MIME al datelor returnate.
COST	Acest câmp nu este implementat în mod curent.
DATE	Conține data creării documentului care se returnează.
DERIVED-FROM	Acest câmp nu este implementat în mod curent.
EXPIRES	Specifică data la care documentul returnat expiră.
LAST-MODIFIED	Specifică data când documentul care se returnează a fost ultima oară modificat.
LINK	Conține informații, precum adresa URL, asupra documentului care este returnat.

(continuare)

Tabelul 25.2. – Continuare

Câmpul antetului de răspuns	*Semnificație*
MESSAGE-ID	Un număr unic de identificare pentru conexiunea HTTP.
PUBLIC	Conține metodele permise pentru cerere. Este similar câmpului ALLOW, dar se aplică tuturor browserelor, nu numai aceluia către care este trimis antetul.
TITLE	Specifică titlul documentului care se returnează.
URI	Conține adresa URI (Uniform Resource Identifier) a documentului care se returnează.
VERSION	Acest câmp nu este implementat în mod curent.

Anteturile de răspuns HTTP se concentrează mai mult pe documentul care este returnat. Aceste informații sunt utilizate de browserul Web pentru a afișa corect documentul o dată ce browserul Web a terminat descărcarea sa.

Când serverul Web primește o cerere HTTP de la o aplicație CGI, el execută programul, în loc să returneze conținutul fișierului. Serverul Web trece, apoi, toate informațiile pe care le recepționează în anteturile de cerere HTTP către aplicația CGI prin plasarea datelor în variabile de mediu.

Înainte ca o aplicație CGI să fi terminat execuția, este nevoie ca ea să returneze un răspuns fie către serverul Web, fie direct browserului Web. Dacă rezultatele urmează să fie trimise serverului Web, este necesar ca aplicația CGI să trimită serverului Web doar o directivă de server. Apoi, serverul Web creează un antet de răspuns HTTP și trimite datele și antetul înapoi către browserul utilizatorului. Dacă aplicația CGI trimite datele direct înapoi către browserul utilizatorului, ea trebuie să creeze întâi un antet de răspuns HTTP valid și să preceadă datele cu antetul.

Trimiterea datelor către aplicația CGI

Cea mai mare parte a datelor disponibile aplicației dumneavoastră CGI parvine de la variabile de mediu setate de serverul Web. Serverul Web plasează toate câmpurile antetului de cerere HTTP în variabile de mediu, care sunt numele câmpului precedat de HTTP_. Suplimentar tuturor anteturilor de cerere HTTP, serverul Web setează și alte variabile de mediu CGI, care sunt arătate în Tabelul 25.3.

Tabelul 25.3. Variabile de mediu CGI.

Variabila de mediu	*Semnificație*
AUTH_TYPE	Specifică metoda de autentificare (dacă există) folosită de browserul Web, cum ar fi metoda username/password.
CONTENT_LENGTH	Conține lungimea, în caractere, a datelor furnizate de utilizator, dacă există.
CONTENT_TYPE	Specifică tipul MIME al datelor furnizate de utilizator.
GATEWAY_INTERFACE	Desemnează versiunea de specificație CGI care este utilizată (versiunea curentă este 1.1).
PATH_INFO	Conține informațiile adiționale despre cale atașate adresei URL cerute.

Variabila de mediu	*Semnificație*
PATH_TRANSLATED	Conține calea reală de pe mașina server, obținută prin translatarea căii virtuale, atașate adresei URL.
QUERY_STRING	Conține orice informație atașată adresei URL cu un semn de întrebare.
REMOTE_ADDR	Furnizează adresa IP a mașinii client.
REMOTE_HOST	Conține numele domeniului mașinii client, dacă este disponibil.
REMOTE_IDENT	Stochează numele login (de deschidere a sesiunii de lucru) al utilizatorului, dacă s-a folosit unul pentru autentificare cu serverul Web.
REMOTE_USER	Conține numele utilizatorului de la distanță, așa cum a fost furnizat serverului Web.
REQUEST_METHOD	Specifică metoda de cerere folosită de browser, fie GET, fie POST.
SCRIPT_NAME	Conține calea virtuală și numele fișierului pentru scriptul CGI.
SERVER_NAME	Conține fie numele de domeniu, fie adresa IP a mașinii server Web.
SERVER_PORT	Indică portul care este utilizat de serverul Web.
SERVER_PROTOCOL	Specifică protocolul care este utilizat între serverul Web și browserul Web, de obicei HTTP.
SERVER_SOFTWARE	Conține numele și versiunea software-ului de server Web.

Variabilele de mediu sunt sursa primară de informații disponibilă aplicației dumneavoastră CGI, iar în unele cazuri singura sursă. Singura altă informație care poate fi trimisă aplicației CGI vine de la unele date tip formular furnizate de utilizator. Aceste date tip formular pot fi trimise serverului Web într-unul din două moduri, cu metodele de cerere GET sau POST.

Termenii GET sau POST ar trebui să vă fie, deja, familiari de la crearea formularelor HTML. Ei reprezintă valorile pe care le dați atributului method al etichetei <FORM>. Cât privește aplicația CGI, diferența între aceste două metode de cerere este locația de unde aplicația CGI poate accesa datele formularului. Cu metoda GET, datele formularului sunt plasate în variabila de mediu QUERY_STRING. Metoda POST, pe de altă parte, face ca datele să fie trimise aplicației CGI prin intermediul intrării standard.

Indiferent care metodă de cerere este utilizată, browserul Web plasează toate datele din formular într-un șir lung codificat URL de perechi nume-valoare. O *pereche nume-valoare* reprezintă numele unui element de formular și datele corespunzătoare introduse de către utilizator în acel element de formular. Toate perechile nume-valoare pentru întregul formular sunt alăturate, separate de caracterul ampersand (&) și, apoi, codificate URL. *Codificarea URL* este procesul de înlocuire a unor caractere cu marcaje de rezervare și a altor caractere cu valoarea lor ASCII hexazecimală. Înainte de a putea începe să lucrați cu datele furnizate de utilizator și trimise aplicației dumneavoastră CGI, este nevoie să decodați datele și să despărțiți perechile nume-valoare.

SFAT

La procesarea datelor din formulare, ar trebui să urmați totdeauna aceeași secvență de evenimente: despărțiți perechile nume-valoare, schimbați marcajele de rezervare în valorile lor respective, iar apoi convertiți valorile hexazecimale înapoi în caracterele echivalente lor.

Returnarea datelor din aplicația CGI

După ce aplicația dumneavoastră CGI și-a terminat treaba, ea trebuie să returneze o valoare browserului Web al utilizatorului. Acest proces poate fi efectuat direct, prin crearea unui antet de răspuns HTTP valid și trimiterea antetului și datelor către browserul Web, sau prin trimiterea datelor și a unei directive de server către serverul Web. În ambele metode aplicația trimite rezultatele către ieșirea sa standard.

Când trimiteți o directivă de server către serverul Web, împreună cu rezultatele de returnat browserului utilizatorului, antetul pe care aplicația dumneavoastră CGI este nevoie să îl creeze este numit *antet analizat* (parsed header). Aceasta întrucât serverul Web analizează antetul pe care îl trimiteți, îl transformă într-un antet de răspuns HTTP complet și trimite rezultatele înapoi browserului Web.

La nivel minim, un antet analizat valid poate conține doar o directivă de server, adică una dintre cele trei definite în Tabelul 25.4. În plus față de directiva de server, aplicația CGI poate, de asemenea, să trimită oricare dintre anteturile de răspuns arătate în Tabelul 25.2. Câmpurile de antet de răspuns HTTP trimise de aplicația CGI ca parte a unui antet analizat sunt, apoi, integrate în antetul de răspuns HTTP complet pe care serverul Web îl creează și îl trimite înapoi către browserul Web.

Tabelul 25.4. Directive de server Web.

Directiva	*Semnificație*
`Content-type`	Desemnează tipul MIME al datelor care se returnează.
`Location`	Desemnează adresa URL virtuală sau absolută către care browserul Web trebuie redirectat.
`Status`	Conține un cod de stare HTTP, cum ar fi `404 Not Found`.

Când o aplicație CGI își trimite rezultatele în mod direct înapoi către browserul utilizatorului, ea trebuie să creeze un antet neanalizat pentru a fi returnat cu datele. Un *antet neanalizat* (non-parsed header) reprezintă un antet de răspuns HTTP pe care aplicația dumneavoastră CGI îl creează și îl trimite browserului Web. Deoarece serverul Web nu analizează antetul și nu suplinește câmpurile care lipsesc, antetul neanalizat trebuie să fie complet. Anteturile de răspuns HTTP valide nu trebuie să conțină toate câmpurile de răspuns HTTP arătate în Tabelul 25.2. De exemplu, ceea ce urmează este un antet neanalizat valid:

```
HTTP/1.0 200 OK
Server: Netscape-Communications/3.0
Content-type: text/html
```

Dacă returnați un antet analizat sau neanalizat din aplicația dumneavoastră CGI, trebuie să returnați o linie goală imediat după ce antetul s-a terminat. Acea linie goală este modul în care serverul Web sau browserul Web identifică sfârșitul antetului și începutul datelor returnate. Dacă nu furnizați linia goală, aplicația dumneavoastră CGI va cauza o eroare de server.

Apelarea aplicațiilor CGI

O dată ce ați dezvoltat o aplicație CGI, sunteți gata să o apelați dintr-unul din documentele HTML. Există o varietate de modalități prin care puteți apela aplicația dumneavoastră. Metoda pe care o folosiți depinde de taskul pe care aplicația CGI este destinată să îl desfășoare.

Cel mai banal mod în care aplicațiile CGI sunt apelate este din atributul `action` al etichetei `<FORM>`. Aceste aplicații CGI sunt denumite *rutine de tratare a formularelor (form handlers)* și sunt apelate când formularul este completat de utilizator. Când un formular este completat, datele din formular sunt trimise aplicației dumneavoastră CGI folosind variabila de mediu `QUERY_STRING` sau intrarea standard, în funcție de metoda de cerere utilizată, `GET` sau `POST`.

Aplicațiile CGI pot fi, de asemenea, apelate direct în etichetele `<A>`, `<IMG>` și cele pentru server-side include (incluziune de partea serverului). Când adresa URL pentru o aplicație CGI este atașată atributului `href` al unei etichete `<A>`, ea va fi rechemată ori de câte ori utilizatorul execută clic pe legătura respectivă în browserul Web. Aplicațiile CGI referite în `<IMG>` și în incluziunile server-side, pe de altă parte, sunt executate automat de către serverul Web când utilizatorul cere pagina Web în care aceste legături sunt înglobate.

De exemplu, ați putea folosi eticheta de imagine:

```
<IMG SRC="/cgi-bin/image">
```

Serverul Web va executa aplicația CGI pentru imagine și va afișa rezultatele ca o imagine în browserul Web. Pentru a folosi această formă de referire la aplicația CGI, este nevoie să fiți sigur că aplicația CGI returnează datele binare pentru o imagine.

Incluziunile server-side sunt similare prin aceea că sunt executate automat de către serverul Web atunci când pagina Web este cerută. Incluziunile server-side au această sintaxă generală:

```
<!--#command tag1="value1" tag2="value2" -->
```

Pentru aplicații CGI, trebuie să folosiți comanda `#exec` pentru ca serverul Web să vă execute aplicația CGI. Apoi specificați incluziunea numelui aplicației CGI folosind atributul `cgi`. De exemplu, aruncați o privire la această incluziune server-side:

```
<!--#exec cgi="/cgi-bin/run_me.pl" -->
```

Ea face serverul Web să ruleze scriptul CGI `run_me.pl`, înglobând rezultatele returnate de la scriptul CGI în documentul HTML unde a fost plasată incluziunea server-side.

Un ultim mod prin care puteți apela aplicațiile dumneavoastră CGI este din scripturile de partea clientului. Scripturile de partea clientului pot cere oricare document care posedă o adresă URL. Primind o cerere de script de partea clientului, o aplicație CGI este asemănătoare folosirii etichetei `<A>` menționate mai înainte; în ambele metode, browserul Web va trimite o cerere HTTP simplă pentru aplicația CGI, făcând ca serverul Web să execute aplicația. Totuși, avantajul utilizării unui script de partea clientului față de o simplă etichetă de ancoră este că scriptul dumneavoastră de partea clientului controlează când aplicația CGI face cererea. Cu eticheta `<A>`, trebuie să așteptați ca utilizatorul să execute clic pe legătură.

Pentru ce este bună CGI?

CGI este bună pentru a vă permite să scrieți aplicații de partea serverului, care extind funcționalitatea serverului dumneavoastră Web. Folosind o aplicație de partea serverului, obțineți mai multe avantaje, precum suportul multiplatformă și multibrowser, mai multe taskuri pe care le puteți completa, putere de procesare crescută și integritatea codului.

După cum, probabil, ați remarcat deja, fiecare browser pe fiecare platformă afișează paginile HTML ușor diferit. Nu toate browserele sunt la fel; ele acceptă versiuni diferite de HTML, programe plug-in și limbaje de scriptare de partea clientului. Natural, puteți crea pagini HTML generice, care lucrează bine pe toate platformele și cu toate browserele. Totuși, ați putea dori să aveți pagini Web interactive, dinamice. Scriptarea de partea clientului vă oferă câteva dintre aceste opțiuni, precum unele de HTML avansat, dar plătiți prețul când se ajunge la suportul pentru toate browserele și mașinile.

Aplicațiile de partea serverului nu au această problemă, atâta vreme cât returnează rezultatele lor în cod HTML afișabil. Desigur, aveți încă unele dintre caracteristicile HTML și de HTML avansat, dar puteți dezvolta aplicații complexe, care generează dinamic pagini Web personale pentru utilizatorii dumneavoastră. Nu trebuie să vă intereseze dacă aplicația dumneavoastră este compatibilă cu browserele lor deoarece ea rulează pe mașina server.

De asemena, de partea serverului aveți mai multe opțiuni de taskuri pe care le puteți îndeplini. Multe resurse care se obișnuiește să fie integrate în situri Web sunt resurse de partea serverului, cum sunt bazele de date. Puteți face uz și de capacitățile limbajelor de programare de nivel mai înalt, care pot rula pe mașina server, dar nu în browserul Web al utilizatorului.

Dispuneți și de puterea de procesare suplimentară a mașinii server Web. Cu toate că mașinile de partea clientului devin mai puternice, mașinile server sunt, încă, mult mai rapide decât copiile lor client. Această putere suplimentară vă permite să dezvoltați taskuri cu folosirea intensivă a procesorului, precum căutări și sortări complexe, fără să degradați prea mult randamentul sitului dumneavoastră.

În sfârșit, prin crearea aplicațiilor de partea serverului, nu trebuie să fiți îngrijorat de expunerea codului sursă al dumneavoastră. Toate limbajele de scriptare de partea clientului furnizează codul sursă pentru beneficiarul final chiar în fereastra browserului. Nu se cere nici un efort special sau talent pentru a-l obține. Chiar miniaplicațiile Java și controalele ActiveX sunt întrucâtva vulnerabile, deoarece utilizatorii dumneavoastră dețin executabilul exact pe mașinile lor. Cei mai talentați pot decompila aceste programe pentru a obține codul sursă.

Aplicațiile de partea serverului, pe de altă parte, domiciliază pe server tot timpul. Fie că sunt programate într-un limbaj de scriptare sau compilate într-un executabil, codul sursă nu ajunge niciodată să fie trimis browserului utilizatorului. În schimb, mașina server execută întotdeauna acest program, când recepționează o cerere pentru el, și returnează numai rezultatele aplicațiilor.

Când se utilizează CGI?

Acum, când cunoașteți unele dintre lucrurile fundamentele despre aplicații CGI și avantajele de a le folosi, v-ați putea întreba când să le folosiți. Răspunsul simplu este când aveți nevoie. Aplicațiile CGI oferă un belșug de putere pentru a adăuga noi trăsături sitului dumneavoastră. Totuși, avantajele se fac simțite numai atunci când aveți nevoie de o caracteristică dată.

Această secțiune ia în discuție mai multe implementări comune de aplicații CGI. Unele dintre ele ar putea fi taskuri de care aveți nevoie pe situl dumneavoastră, dar altele pot să nu vă aducă vreun avantaj. Dumneavoastră trebuie să determinați ce aveți nevoie să înfăptuiți și apoi să implementați în mod adecvat.

Motoare de căutare

Motoarele de căutare sunt aplicații care permit utilizatorilor să caute prin documentele de pe situl dumneavoastră Web cuvinte cheie sau fraze. Motoarele de căutare pot fi relativ simple sau deosebit de complexe, depinzând de caracteristicile pe care doriți să le accepte. De exemplu, o căutare după un cuvânt cheie este mult mai complicată decât căutarea folosind opțiuni predefinite.

Motoarele de căutare necesită interfața CGI pentru a funcționa corect. Ele trebuie să fie capabile să primească date de intrare, precum un cuvânt cheie pe care să îl caute, din partea

browserului Web al utilizatorului și apoi să returneze rezultatele căutării către browser pentru afișare.

Chiar și un motor de căutare minimal poate constitui o solicitare uriașă pentru procesorul unei mașini. O căutare minimală constă din două dintre taskurile cele mai solicitante pentru procesor pe care o aplicație le poate înfăptui: căutarea și sortarea. De aceea, un motor de căutare Web trebuie să fie optimizat pentru a oferi nivelul de performanță pe care utilizatorii dumneavoastră îl vor dori.

Dezvoltarea propriului dumneavoastră motor de căutare poate fi o sarcină incomodă, chiar pentru o căutare simplă. Exceptând cazul în care vreți să vă încercați puterile sau să dezvoltați un nivel mai bun de tehnologie, ar fi bine să o evitați. O opțiune mai potrivită ar fi să obțineți licența pentru tehnologia de căutare de la o companie specializată în crearea acestui tip de software. Puteți, apoi, să vă petreceți timpul dezvoltând alte zone ale sitului dumneavoastră Web.

Unul dintre cele mai bune motoare de căutare oferite cu licență pentru situri Web este Excite for Web Servers al firmei Excite. El acceptă căutări după cuvinte cheie complexe și afișează rezultatele într-un format ierarhizat. Versiunea actuală posedă chiar o nouă tehnologie, care returnează documente asociate care nici măcar nu conțin cuvintele cheie ale utilizatorului.

Excite oferă o licență gratuită de Excite for Web Servers. Termenii de acordare a licenței reclamă anumite mențiuni ale numelui și emblemei Excite în situl dumneavoastră Web. Puteți găsi convenția curentă de acordare a licenței și să descărcați ultima versiune de pe situl Web Excite (`http://www.excite.com/navigate/`).

Numărătoare de acces

Numărătoarele de acces constituie o trăsătură populară a multor situri Web. Ele utilizează o aplicație CGI care numără de câte ori o pagină Web a fost cerută și, apoi, afișează acest total undeva pe pagină într-un format de tip text sau grafic.

Numărătoarele de acces bazate pe text simplu nu sunt dificil de implementat. De fapt, cea mai mare parte a codului cerut este pentru a realiza taskuri de deschidere a fișierului pentru numărător (care stochează numărul curent de cereri ale paginii Web), de citire a conținutului, incrementare a numărului și stocare a sa înapoi în fișier. Dincolo de acestea, aplicația CGI este necesar doar să afișeze numărul curent pe pagina Web.

Numărătoarele de acces grafice cer, natural, ceva mai mult cod. Este necesar de asemenea să deschideți fișierul pentru numărător și să incrementați numărul de cereri, dar trebuie să creați și un grafic de reprezentare a numărului de cereri pentru pagina Web. Aplicația CGI afișează deci graficul, nu doar un simplu text, în pagina Web.

Codul Perl din Listingul 25.1 conține o versiune de script CGI a unui numărător de acces bazat pe text. Acest cod posedă o subrutină, `Increment_Counter`, care realizează următoarele taskuri:

- Deschide fișierul de numărător pentru introducere
- Citește conținutul
- Închide fișierul de numărător
- Incrementează numărul
- Deschide fișierul de numărător pentru operația de ieșire
- Stochează valoarea numărului incrementat în fișierul de numărător
- Închide fișierul de numărător

Listingul 25.1. Un script numărător de acces bazat pe text.

```
#!/usr/bin/perl

$file = "/users/robertm/count.dat";

$access_number = &Increment_Counter;
print "Content-type: text/html\n\n";
print $access_number;

Sub &Increment_Counter {
  local ($count);

  # Get the current value of the access counter.
  open(COUNT, "$file") || die "Content-type: text/html\n\nCannot open counter
  ➥file!";
$count = <COUNT>;
  close(COUNT);

  # increment the access counter
  $count++;

  # Store the value of the counter in the counter file.
  open(COUNT, ">$file") || die "Content-type: text/html\n\nCannot open counter
  ➥file!";
  print COUNT $count;
  close(COUNT);

  return #count;
}
```

Cu subrutina `Increment_Counter`, tot ceea ce mai este necesar să facă scriptul este să apeleze subrutina și să trimită valoarea returnată browserului Web. Aceasta se face prin liniile următoare de cod Perl:

```
$access_number = &Increment;
print "Content-type: text/html\n\n";
print $access_number;
```

Acum, că aveți scriptul dumneavoastră CGI pentru numărător de acces, sunteți gata să îl implementați. Primul lucru pe care îl aveți de făcut este să creați fișierul `count.dat`. Asigurați-vă că așezați acest fișier în directorul pe care l-ați atribuit variabilei `$file` din scriptul dumneavoastră. Altminteri, scriptul nu va fi capabil să găsească fișierul. Dacă doriți să porniți numărătorul de la 1, este nevoie să puneți un 0 în fișier deoarece valoarea este totdeauna incrementată înainte de a fi afișată în browserul Web.

Cu scriptul CGI terminat și fișierul pentru numărător inițializat, sunteți pregătit să adăugați numărătorul paginii dumneavoastră HTML. Cel mai bun mod de a face aceasta este printr-o incluziune server-side, cum se arată în continuare:

```
<!--#exec cgi="/cgi-bin/counter.pl" -->
```

Pur și simplu plasați această linie de cod HTML la locul exact din pagina dumneavoastră HTML unde doriți să apară numărul. De exemplu, fișierul HTML arătat în Listingul 25.2 conține această incluziune server-side. Figura 25.1 arată cum se vede această pagină când este încărcată în Netscape.

Listingul 25.2. Fișierul HTML pentru numărător de acces bazat pe text.

```
<HTML>
<HEAD>
<TITLE>Including a Text Access Counter</TITLE>
</HEAD>
<BODY BGCOLOR="#CCFFCC">
<TABLE BORDER=0 WIDTH=100% HEIGHT=100%>
<TR>
<TD VALIGN=center ALIGN=middle><FONT FACE="COURIER" SIZE=+1>thank you
for visiting our site!<B>you are visitor number</FONT><FONT FACE="TIMES"
➥ SIZE=+2><B>
<!--#exec cgi="/cgi-bin/counter.pl" -->
</B></FONT></TD>
</TR>
</TABLE>
</BODY>
</HTML>
```

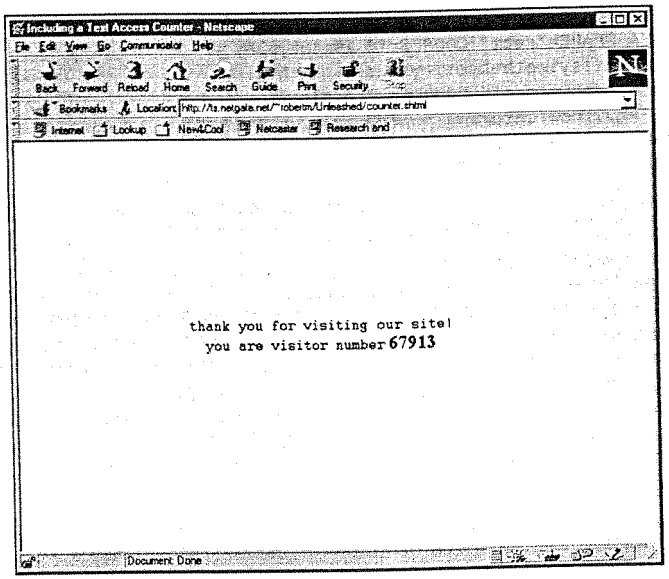



FIGURA 25.1.
Un numărător de acces bazat pe text.

Rutine de tratare a formularelor

Pentru aplicațiile CGI, manevrarea formularelor este cel mai obișnuit task de realizat. Folosind formulare HTML în situl dumneavoastră Web, puteți obține o reacție inversă direct de la utilizatorii dumneavoastră. Însă, dacă nu doriți să primiți toate formularele prin e-mail, aveți nevoie de o aplicație CGI care să trateze expedierea formularelor.

De exemplu, codul HTML din Listingul 25.3 conține un formular de pagină Web care este folosit pe situl Web unGROOM'd (`http://www.ungroomd.com`) pentru a obține date demografice despre utilizatorii săi. Figura 25.2 arată cum se vede această pagină în Netscape Navigator.

Listingul 25.3. Studiul unKNOW'n.

```
<HTML>
<HEAD>
```

(continuare)

Listingul 25.3. – Continuare

```
<TITLE>unGROOM'd: unKNOW'n</TITLE>
</HEAD>
<BODY BACKGROUND="../background2.gif" BGCOLOR="#000000" TEXT="#FFFFCC"
➥ LINK="#FFA500" VLINK="#F7007B">
<TABLE BORDER="0" CELLPADDING="0" WIDTH="100%">
<TR>
<TD ALIGN=LEFT VALIGN=TOP>

<IMG SRC="../1pixel.gif" ALIGN=TOP WIDTH="165" HEIGHT="75" BORDER="0"><BR>
<MAP NAME="upperbar">
<AREA SHAPE="rect" COORDS="-6, 178, 133, 201" HREF="../../unarchivd.html">
<AREA SHAPE="rect" COORDS="6,127,135,178" HREF="../columns/columns.html">
<AREA SHAPE="rect" COORDS="6,103,135,127" HREF="../hiscastle/hiscastle.html">
<AREA SHAPE="rect" COORDS="6,75,135,103" HREF="../incout/incout.html">
<AREA SHAPE="rect" COORDS="6,53,135,75" HREF="../mom/mom.html">
<AREA SHAPE="rect" COORDS="6,29,135,53"
HREF="../tyingtheknot/tyingtheknot.html">
<AREA SHAPE="rect" COORDS="6,2,135,29"
HREF="../nowfeaturing/nowfeaturing.html">
<AREA SHAPE=DEFAULT HREF="../toc.html">
</MAP>

<IMG SRC="../upperbar.gif" ALT="navigation bar" ALIGN=TOP WIDTH="135"
HEIGHT="229" BORDER="0" USEMAP="#upperbar"><BR>

<MAP NAME="level1bar">
<AREA SHAPE="rect" COORDS="6, 146, 131, 174" HREF="mmg_pr/pr.html">
<AREA SHAPE="rect" COORDS="6,124,135,147" HREF="mmg_pr/mmg.html">
<AREA SHAPE="rect" COORDS="6,99,135,125" HREF="staff/credits.html">
<AREA SHAPE="rect" COORDS="6,73,135,99" HREF="unlinkd.html">
<AREA SHAPE="rect" COORDS="6,52,135,74" NOHREF>
<AREA SHAPE="rect" COORDS="6,28,135,53" HREF="unlistd/unlistd.html">
<AREA SHAPE="rect" COORDS="6,3,135,28" HREF="uneditd/unedited.html">
<AREA SHAPE=DEFAULT HREF="../toc.html">
</MAP>

<IMG SRC="../level1bar.gif" Alt="alternate navigation bar" ALIGN=top
WIDTH="135" HEIGHT="208" BORDER="0" USEMAP="#level1bar">

</TD>

<TD ALIGN=LEFT VALIGN=TOP>

<TABLE BORDER="0" CELLPADDING="0" WIDTH="400">
<TR>
<TD ALIGN=LEFT VALIGN=TOP >
<IMG SRC="unknown.gif" ALIGN=MIDDLE WIDTH="200" HEIGHT="50" BORDER="0"><BR><BR>
</TD>
<TD ALIGN=LEFT VALIGN=TOP ></TD>
<TD ALIGN=RIGHT VALIGN=TOP >
<A HREF="../toc.html"><IMG SRC="../150plogo.gif" ALIGN=TOP WIDTH="150"
HEIGHT="63" BORDER="0"></A><SMALL><SUP>TM</SUP></SMALL></TD>
</TR>
<TR>
<TD ALIGN=LEFT VALIGN=TOP COLSPAN="3"><BR><HR>
<IMG SRC="../1pixel.gif" ALIGN=TOP WIDTH="400" HEIGHT="1" BORDER="0">
```



```
<B>unGROOM'd<SMALL><SUP>TM</SUP></SMALL></B> is dedicated to providing men with
an on line service to help with your "engagement, marriage, and what
follows....." Please take the time to fill out our survey so we can better
understand our readers and what topics are most important to you. We thank you
for your time and support!<BR>
<P><HR>
</TD>
</TR>
<TR>
<TD ALIGN=LEFT VALIGN=TOP COLSPAN="3">
<A NAME="top"></A>
<FORM ACTION="/ungroomd/cgi-bin/survey.pl" METHOD="POST">
<P><INPUT TYPE="text" NAME="name" VALUE="" SIZE=40>Name
<P><INPUT TYPE="text" NAME="email" VALUE="" SIZE=30>E-mail address
<P><B>Are You:</B>
<P><B>Married?</B> <INPUT TYPE="radio" NAME="married" VALUE="Yes">Yes
<INPUT TYPE="radio" NAME="married" VALUE="No">No
<P>If yes, how many years? Years:<INPUT TYPE="text" NAME="yearsmarried"
VALUE="" SIZE=3 MAXLENGTH=3>
<P><B>In a relationship?</B> <INPUT TYPE="radio" NAME="relationship"
VALUE="Yes">Yes <INPUT TYPE="radio" NAME="relationship" VALUE="No">No
<P>If yes, how many years? Years:<INPUT TYPE="text" NAME="yearsrelationship"
VALUE="" SIZE=3 MAXLENGTH=3>
<P><B>Are your parents still married?</B>
<P><INPUT TYPE="radio" NAME="parents" VALUE="yes">Yes
<INPUT TYPE="radio" NAME="parents" VALUE="no">No
<P><B>Would you rather?</B>
<P><INPUT TYPE="radio" NAME="howmarried" VALUE="elope">Elope
<INPUT TYPE="radio" NAME="howmarried" VALUE="have wedding">Have Wedding
<P><B>Do you read any books on marriage/relationships?</B>
<P><INPUT TYPE="radio" NAME="readbooks" VALUE="yes">Yes
<INPUT TYPE="radio" NAME="readbooks" VALUE="no">No
<P>If no, any interest in them? <INPUT TYPE="radio" NAME="booksinterest"
VALUE="yes">Yes <INPUT TYPE="radio" NAME="booksinterest" VALUE="no">No
<P><B>Have you ever cheated sexually on your partner?</B>
<P><INPUT TYPE="radio" NAME="cheated" VALUE="yes">Yes
<INPUT TYPE="radio" NAME="cheated" VALUE="no">No
<P><B>For those who have taken the plunge:</B>
<P><B>How long were you engaged for?</B>
<P><INPUT TYPE="radio" NAME="timeengaged" VALUE="less than a year">Less than a
year <INPUT TYPE="radio" NAME="timeengaged" VALUE="1-2">1-2 Years <INPUT
TYPE="radio" NAME="timeengaged" VALUE="2+">2+ Years
<P><B>Did you feel part of the wedding planning process?</B>
<P><INPUT TYPE="radio" NAME="planning" VALUE="yes">Yes
<INPUT TYPE="radio" NAME="planning" VALUE="no">No
<P><B>Did you attend bridal events?</B>
<P><INPUT TYPE="radio" NAME="bridalevents" VALUE="yes">Yes
<INPUT TYPE="radio" NAME="bridalevents" VALUE="no">No
<P><B>How much did you spend on the engagement ring?</B>
<P><INPUT TYPE="radio" NAME="ring" VALUE="1-3">$1,000 -3,000
<INPUT TYPE="radio" NAME="ring" VALUE="3-5">$3,000 -5,000
<INPUT TYPE="radio" NAME="ring" VALUE="5-8"> $5,000 - 10,000
<INPUT TYPE="radio" NAME="ring" VALUE="10+">$10,000 +
```

(continuare)

Listingul 25.3. – Continuare

```
<P><B>How much did you/are you spending on your wedding?</B>
<P><INPUT TYPE="radio" NAME="weddingcost" VALUE="1-5">$1,000-5,000
<INPUT TYPE="radio" NAME="weddingcost" VALUE="5-8">$5,000-10,000
<INPUT TYPE="radio" NAME="weddingcost" VALUE="10-15">$10,000-15,000
<INPUT TYPE="radio" NAME="weddingcost" VALUE="15-25">$15,000 - 25,000
<INPUT TYPE="radio" NAME="weddingcost" VALUE="25+">$25,000+
<P><B>Who organized the Honeymoon Details?</B>
<P><INPUT TYPE="radio" NAME="honeymoondetails" VALUE="her">Her
<INPUT TYPE="radio" NAME="honeymoondetails" VALUE="you">You
<INPUT TYPE="radio" NAME="honeymoondetails" VALUE="both">Both
<P><B>Did you see a travel agent?</B>
<P><INPUT TYPE="radio" NAME="travelagent" VALUE="yes">Yes
<INPUT TYPE="radio" NAME="travelagent" VALUE="no">No
<P><B>How long was your honeymoon?</B>
<P><INPUT TYPE="radio" NAME="honeymoonlength" VALUE="1 week">1 week or less
<INPUT TYPE="radio" NAME="honeymoonlength" VALUE="2-3 weeks">2-3 Week
<INPUT TYPE="radio" NAME="honeymoonlength" VALUE="4 weeks+">4 weeks or more
<INPUT TYPE="radio" NAME="honeymoonlength" VALUE="none">No honeymoon
<P><B>How much was your honeymoon?</B>
<P><INPUT TYPE="radio" NAME="honeymooncost" VALUE="less than 1000">$1,000 or
less
<INPUT TYPE="radio" NAME="honeymooncost" VALUE="2,000-4,000">$2,000 - 4,000
<INPUT TYPE="radio" NAME="honeymooncost" VALUE="4,000-6,000">$4,000-6,000
<INPUT TYPE="radio" NAME="honeymooncost" VALUE="7,000">$6,000 or more
<P><B>Did you have any marriage counseling prior to marriage or now that you
are married?</B>
<P><INPUT TYPE="radio" NAME="counseling" VALUE="yes">Yes
<INPUT TYPE="radio" NAME="counseling" VALUE="no">No
<P><B>A little about yourself:</B>
<P><B>Age:</B><INPUT TYPE="radio" NAME="age" VALUE="18-24">18-24
<INPUT TYPE="radio" NAME="age" VALUE="25-30">25-30
<INPUT TYPE="radio" NAME="age" VALUE="31-35">31-35
<INPUT TYPE="radio" NAME="age" VALUE="36-40">36-40
<INPUT TYPE="radio" NAME="age" VALUE="40+" CHECKED>40+
<P><B>Sex: </B><INPUT TYPE="radio" NAME="sex" VALUE="male">Male
<INPUT TYPE="radio" NAME="sex" VALUE="female">Female
<P><B>Estimated yearly income?</B>
<INPUT TYPE="radio" NAME="income" VALUE="0-20">0-$20,000
<INPUT TYPE="radio" NAME="income" VALUE="20-40">$20,000-$40,000
<INPUT TYPE="radio" NAME="income" VALUE="40-60">$40,000-60,000
<INPUT TYPE="radio" NAME="income" VALUE="60-100">$60,000-$100,000
<INPUT TYPE="radio" NAME="income" VALUE="100+">$100,000+
<P><B>Education:</B>
<INPUT TYPE="radio" NAME="education" VALUE="High School">High School
<INPUT TYPE="radio" NAME="education" VALUE="Some College">Some College
<INPUT TYPE="radio" NAME="education" VALUE="College Degree">College Degree
<INPUT TYPE="radio" NAME="education" VALUE="Graduate Degree">Graduate Degree
<INPUT TYPE="radio" NAME="education" VALUE="other">Other
<P><B>Weekly internet usage:</B>
<INPUT TYPE="radio" NAME="internet" VALUE="less than an hour">1 Hour or less
<INPUT TYPE="radio" NAME="internet" VALUE="2-3">2-3 Hours
<INPUT TYPE="radio" NAME="internet" VALUE="3-5">3-5 Hours
```



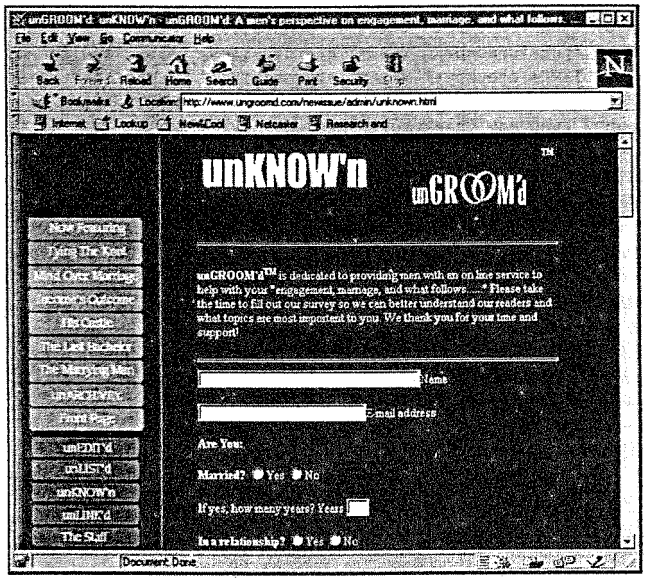
```
<INPUT TYPE="radio" NAME="internet" VALUE="5+">5 Hours or more
<P><B>What do you use the web for?</B>
<INPUT TYPE="checkbox" NAME="webusage" VALUE="business">Business
<INPUT TYPE="checkbox" NAME="webusage" VALUE="Entertainment">Entertainment
<INPUT TYPE="checkbox" NAME="webusage" VALUE="Research">Research
<P><B>Favorite types of magazines (you may check more than one box):</B>
<P><INPUT TYPE="checkbox" NAME="magazines" VALUE="music">Music
<INPUT TYPE="checkbox" NAME="magazines" VALUE="entertainment">Entertainment
<INPUT TYPE="checkbox" NAME="magazines" VALUE="business">Business
<INPUT TYPE="checkbox" NAME="magazines" VALUE="adult entertainment">Adult
Entertainment
<INPUT TYPE="checkbox" NAME="magazines" VALUE="sports">Sports
<INPUT TYPE="checkbox" NAME="magazines" VALUE="mens">Men's Publications
<INPUT TYPE="checkbox" NAME="magazines" VALUE="financial">Financial
<INPUT TYPE="checkbox" NAME="magazines" VALUE="computer">Computer
<P><B>Favorite types of Internet sites (you may check more than one box):</B>
<P><INPUT TYPE="checkbox" NAME="internetsites" VALUE="music">Music
<INPUT TYPE="checkbox" NAME="internetsites" VALUE="entertainment">Entertainment
<INPUT TYPE="checkbox" NAME="internetsites" VALUE="business">Business
<INPUT TYPE="checkbox" NAME="internetsites" VALUE="adult">Adult Entertainment
<INPUT TYPE="checkbox" NAME="internetsites" VALUE="sports">Sports
<P><B>What section of unGROOM'd<SMALL><SUP>TM</SUP></SMALL> is of most interest
to you? (you may check more than one box):</B>
<P><INPUT TYPE="checkbox" NAME="ungroomdsection" VALUE="now featuring">Now
Featuring
<INPUT TYPE="checkbox" NAME="ungroomdsection" VALUE="tying the knot">Tying the
Knot
<INPUT TYPE="checkbox" NAME="ungroomdsection" VALUE="mind over marriage">Mind
Over Marriage
<INPUT TYPE="checkbox" NAME="ungroomdsection" VALUE="income's outcome">
Income's Outcome
<INPUT TYPE="checkbox" NAME="ungroomdsection" VALUE="his castle">His Castle
<INPUT TYPE="checkbox" NAME="ungroomdsection" VALUE="last bachelor/MM">The Last
Bachelor/The Marrying Man
<P><B>What topics related to marriage would you like to see covered in
unGROOM'd<SMALL><SUP>TM</SUP></SMALL>? (you may check more than one box):</B>
<P><INPUT TYPE="checkbox" NAME="topics" VALUE="finacial">Financial Issues
<INPUT TYPE="checkbox" NAME="topics" VALUE="Psychology">Psychology Issues
<INPUT TYPE="checkbox" NAME="topics" VALUE="Wedding Day">Wedding Day Issues
<INPUT TYPE="checkbox" NAME="topics" VALUE="Becoming a father">Becoming a
Father Issues
<INPUT TYPE="checkbox" NAME="topics" VALUE="Infidelity">Infidelity Issues
<INPUT TYPE="checkbox" NAME="topics" VALUE="Religious">Religious Issues
<P><B>How did you hear about unGROOM'd<SMALL><SUP>TM</SUP></SMALL>?</B>
<P><INPUT TYPE="checkbox" NAME="hearaboutus" VALUE="internet search">Internet
Search
<INPUT TYPE="checkbox" NAME="hearaboutus" VALUE="Followed a link"> Followed a
link
<INPUT TYPE="checkbox" NAME="hearaboutus" VALUE="Print article">Print article
<INPUT TYPE="checkbox" NAME="hearaboutus" VALUE="Word of mouth">Word of mouth
<INPUT TYPE="checkbox" NAME="hearaboutus" VALUE="Radio">Radio
<INPUT TYPE="checkbox" NAME="hearaboutus" VALUE="Advertisement">Advertisement
```

(continuare)

Listingul 25.3. – Continuare

```
<INPUT TYPE="checkbox" NAME="hearaboutus" VALUE="Flyer/hand out">Flyer/hand out
<P><B>Any Additional Comments?</B>
<TEXTAREA NAME="comments" ROWS=4 COLS=40></TEXTAREA>
<P><INPUT TYPE="submit" NAME="Submit" VALUE="Submit"><INPUT TYPE="reset"
VALUE="Reset">
<P><CENTER><B>THANK YOU FOR YOUR TIME &amp;amp; INPUT TO
unGROOM'd<SMALL><SUP>TM</SUP></SMALL>!!!</B>
</CENTER>
</FORM>
</TD>
</TR>
</TABLE>
</TD></TR></TABLE>
</BODY>
</HTML>
```

FIGURA 25.2.
Studiul unKNOW'n.



Observați această linie din Listingul 25.3:

```
<FORM ACTION="/ungroomd/cgi-bin/survey.pl" METHOD="POST">
```

Aceasta este eticheta `<FORM>`, care specifică aplicația CGI de apelat când formularul este completat.

Toate aplicațiile CGI de tratare a formularelor trebuie să efectueze două taskuri. Primul task este să recepționeze și să decodifice datele din formular trimise de către browserul Web. Al doilea task variază puternic, depinzând de anvergura aplicației dumneavoastră CGI. De exemplu, ați putea dori să plasați datele într-o bază de date pentru acces ușor. În acest caz, al doilea task pe care aplicația dumneavoastră CGI îl va înfăptui este să plaseze datele în baza de date.

Această secțiune se concentrează asupra primului task – recepționarea și decodificarea datelor de formular – pentru scriptul CGI `survey.pl`. Al doilea task este explicat în secțiunea următoare, „Integrarea bazelor de date".

Deoarece browserul Web codifică URL toate datele de formular înainte de a le trimite serverului Web, scriptul dumneavoastră CGI trebuie să decodifice datele înainte să începeți să lucrați cu ele. Subrutina Perl arătată în Listingul 25.4 citește datele de formular ale utilizatorului și, apoi, elimină toate codificările URL.

Listingul 25.4. Subrutina `User_Data`.

```
sub User_Data {
  local (%user_data, $user_string, $name_value_pair,
        @name_value_pairs, $name, $value);

  # If the data was sent via POST, then it is available
  # from standard input. Otherwise, the data is in the
  # QUERY_STRING environment variable.
  if ($ENV{'REQUEST_METHOD'} eq "POST") {
    read(STDIN,$user_string,$ENV{'CONTENT_LENGTH'});
  } else {
    $user_string = $ENV{'QUERY_STRING'};
  }

  # This line changes the + signs to spaces.
  $user_string =~ s/\+/ /g;

  # This line places each name/value pair as a separate
  # element in the name_value_pairs array.
  @name_value_pairs = split(/&/, $user_string);

  # This code loops over each element in the name_value_pairs
  # array, splits it on the = sign, and places the value
  # into the user_data associative array with the name as the
  # key.
  foreach $name_value_pair (@name_value_pairs) {
    ($name, $value) = split(/=/, $name_value_pair);

    # These two lines decode the values from any URL
    # hexadecimal encoding. The first section searches for a
    # hexadecimal number and the second part converts the
    # hex number to decimal and returns the character
    # equivalent.
    $name =~
    s/%([a-fA-F0-9][a-fA-F0-9])/pack("C",hex($1))/ge;
    $value =~
      s/%([a-fA-F0-9][a-fA-F0-9])/pack("C",hex($1))/ge;

    # If the name/value pair has already been given a value,
    # as in the case of multiple items being selected, then
    # separate the items with a ":".
    if (defined($user_data{$name})) {
      $user_data{$name} .= ":" . $value;
    } else {
      $user_data{$name} = $value;
    }
  }
  return %user_data;
}
```

Observați ordinea taskurilor în Listingul 25.4. Mai întâi, se obțin datele utilizatorului fie de la variabila de mediu QUERY_STRING, fie de la intrarea standard, depinzând de ce metodă de cerere a fost folosită. Apoi, toate semnele plus sunt reconvertite în spații, urmate de despărțirea perechilor nume-valoare.

Ceea ce rămâne din subrutină este o buclă ce efectuează o iterație pentru fiecare pereche nume-valoare. O dată aflate în buclă, perechile nume-valoare sunt despărțite și plasate în variabile separate. Apoi, toate numerele hexazecimale sunt modificate înapoi în echivalenții lor tip caracter. În final, numele și valoarea sunt plasate într-un tablou asociativ, cu porțiunea nume acționând ca index al elementului de tablou. În exemplu, tabloul asociativ este %user_data. Valoarea pe care utilizatorul a introdus-o în câmpul de intrare Email din formular va fi stocată în elementul de tablou $user_data{"email"}.

Observați că, dacă există deja un index în tabloul asociativ cu același nume, semnul ; (punct și virgulă) și noua valoare sunt atașate elementului de tablou existent. Această adăugare este făcută când se pot trimite opțiuni multiple de către utilizator, cum se întâmplă cu casetele de validare și elementele de selecție.

Subrutina User_Data decodifică datele din formularul utilizatorului și plasează rezultatele într-un tablou asociativ. După decodificarea datelor, este o idee bună să eliminați orice incluziune server-side pe care utilizatorul este posibil să o fi introdus. Tipic, singurii utilizatori care introduc incluziuni server-side fac astfel pentru a încerca să compromită securitatea mașinii dumneavoastră, drept care ar trebui să verificați și să eliminați respectivele introduceri. Subrutina No_SSI, arătată în Listingul 25.5, realizează acest task.

Listingul 25.5. Subrutina No_SSI.

```
sub No_SSI {
  local (*data) = @_;

  foreach $key (sort keys(%data)) {
      $data{$key} =~ s/<!--(.|\n)*-->//g;
  }
}
```

În secțiunea următoare, dezvoltați restul scriptului CGI survey.pl pentru a trata formularul unKNOW'n Survey. Când faceți retușurile finale asupra fișierului survey.pl, cele două subrutine pe care le-ați dezvoltat în această secțiune vor fi integrate în el.

Integrarea bazelor de date

Integrarea sitului dumneavoastră Web cu o bază de date este o altă aplicație populară a CGI. Procedând astfel, puteți stoca multe din datele dumneavoastră într-o bază de date și regenera paginile dumneavoastră Web ori de câte ori datele se modifică, prin aceasta păstrându-vă paginile Web la zi. O altă utilizare comună a unei baze de date este să stocheze răspunsurile pe care le-ați obținut de la utilizatorii care au completat formularele de pe situl dumneavoastră Web. Puteți, apoi, să generați cu ușurință rapoarte la zi din toate răspunsurile pe care le-ați primit.

Multe companii mari folosesc într-un fel baze de date cu siturile lor Web. Aceste baze de date sunt, adesea, sisteme comerciale complete de administrare a bazelor de date relaționale, cum ar fi Oracle, Informix sau Sybase. Pentru baze de date cu o mulțime de înregistrări, aceste sisteme sunt soluțiile preferate.

Totuși, pentru proiecte mai mici, bazele de date de tip fișier text sunt adesea suficiente pentru necesitățile sitului Web. Aceste baze de date sunt simple fișiere tip text, cu fiecare

înregistrare pe o linie separată din fișier. Câmpurile din fiecare înregistrare sunt delimitate de un anume caracter, precum virgulă sau punct și virgulă. Puteți avea, de asemenea, înregistrări singulare care să ocupe mai multe linii din fișier și să delimitați începutul sau sfârșitul unei înregistrări cu o secvență specială de caractere.

În secțiunea anterioară, ați dezvoltat prima parte a scriptului CGI survey.pl, care va manevra intrările de formular trimise de studiul unKNOW'n, arătat în Listingul 25.3. Acum, că datele din formular au fost decodificate, puteți formata aceste informații și să le stocați într-o bază de date de tip fișier text pentru acces simplu. Codul de adăugare a datelor formatate, împreună cu cele două subrutine dezvoltate în secțiunea anterioară, este arătat în Listingul 25.6.

Listingul 25.6. Fișierul survey.pl.

```
#!/usr/local/bin/perl5
#*******************************************************
# This script maintains a flat-file database of
# ungroomd.com survey responses.
#*******************************************************
$HOME = (getpwnam("www03640"))[7];
$path = "$HOME/survey";
$database_file = ">>$path/unKNOWn.dat";
$quote = "\"";
$ASCII_ten = pack("c", 10);
$ASCII_thirteen = pack("c", 13);
$space = " ";

# Format the Web data received and make sure
# it does not contain any server-side includes
%data_received = &User_Data;
&No_SSI(*data_received);

# Make sure all mandatory fields are full and
# truncate all strings to a maximum of 255 characters
# Also make sure no new-line characters.
foreach $key (keys(%data_received)) {

  $data_received{$key} =~ s/$ASCII_ten//ge;
  $data_received{$key} =~ s/$ASCII_thirteen/$space/ge;

}

open(DATABASE,"$database_file") ||
  die "Content-type: text/html\n\nCannot open database.";

# Lock the database file now.
flock(DATABASE, 2);

print DATABASE $quote . $data_received{"name"} . $quote . "," .
    $quote . $data_received{"email"} . $quote . "," .
    $quote . $data_received{"married"} . $quote . "," .
    $quote . $data_received{"yearsmarried"} . $quote . "," .
    $quote . $data_received{"relationship"} . $quote . "," .
    $quote . $data_received{"yearsrelationship"} . $quote . "," .
    $quote . $data_received{"parents"} . $quote . "," .
    $quote . $data_received{"howmarried"} . $quote . "," .
    $quote . $data_received{"readbooks"} . $quote . "," .
```

(continuare)

Listing 25.6. – Continuare

```
    $quote . $data_received{"booksinterest"} . $quote . "," .
    $quote . $data_received{"cheated"} . $quote . "," .
    $quote . $data_received{"timeengaged"} . $quote . "," .
    $quote . $data_received{"planning"} . $quote . "," .
    $quote . $data_received{"bridalevents"} . $quote . "," .
    $quote . $data_received{"ring"} . $quote . "," .
    $quote . $data_received{"weddingcost"} . $quote . "," .
    $quote . $data_received{"honeymoondetails"} . $quote . "," .
    $quote . $data_received{"travelagent"} . $quote . "," .
    $quote . $data_received{"honeymoonlength"} . $quote . "," .
    $quote . $data_received{"honeymooncost"} . $quote . "," .
    $quote . $data_received{"counseling"} . $quote . "," .
    $quote . $data_received{"age"} . $quote . "," .
    $quote . $data_received{"sex"} . $quote . "," .
    $quote . $data_received{"income"} . $quote . "," .
    $quote . $data_received{"education"} . $quote . "," .
    $quote . $data_received{"internet"} . $quote . "," .
    $quote . $data_received{"webusage"} . $quote . "," .
    $quote . $data_received{"magazines"} . $quote . "," .
    $quote . $data_received{"internetsites"} . $quote . "," .
    $quote . $data_received{"ungroomdsection"} . $quote . "," .
    $quote . $data_received{"topics"} . $quote . "," .
    $quote . $data_received{"hearaboutus"} . $quote . "," .
    $quote . $data_received{"comments"} . $quote . "\n";
flock(DATABASE, 8);
close(DATABASE);

# Output success message
print "Location:http://www.ungroomd.com/newissue/admin/unknown-success.html\n\n";

# User_Data
# This procedure receives form input from a Web
# page form and formats it into an associative
# array. The keys of the associative array are
# the names of the form elements. The associative
# array is returned from the procedure.
sub User_Data {
local (%user_data, $user_string, $name_value_pair,
       @name_value_pairs, $name, $value);

# If the data was sent via POST, then it is available
# from standard input. Otherwise, the data is in the
# QUERY_STRING environment variable.
if ($ENV{'REQUEST_METHOD'} eq "POST") {
  read(STDIN,$user_string,$ENV{'CONTENT_LENGTH'});
} else {
  $user_string = $ENV{'QUERY_STRING'};
}

# This line changes the + signs to spaces.
$user_string =~ s/\+/ /g;

# This line places each name/value pair as a separate
# element in the name_value_pairs array.
@name_value_pairs = split(/&/, $user_string);

# This code loops over each element in the name_value_pairs
# array, splits it on the = sign, and places the value
# into the user_data associative array with the name as the
# key.
foreach $name_value_pair (@name_value_pairs) {
  ($name, $value) = split(/=/, $name_value_pair);

  # These two lines decode the values from any URL
  # hexadecimal encoding. The first section searches for a
  # hexadecimal number and the second part converts the
  # hex number to decimal and returns the character
  # equivalent.
  $name =~
  s/%([a-fA-F0-9][a-fA-F0-9])/pack("C",hex($1))/ge;
  $value =~
    s/%([a-fA-F0-9][a-fA-F0-9])/pack("C",hex($1))/ge;

  # If the name/value pair has already been given a value,
  # as in the case of multiple items being selected, then
  # separate the items with a ":".
  if (defined($user_data{$name})) {
    $user_data{$name} .= ":" . $value;
  } else {
    $user_data{$name} = $value;
  }
}
return %user_data;
}

# No_SSI
# This procedure checks form data entered by
# a user, and removes any SSI (server side
# includes) fields. It receives an associative
# array as the input.
sub No_SSI {
  local (*data) = @_;

  foreach $key (sort keys(%data)) {
    $data{$key} =~ s/<!--(.|\n)*-->//g;
    }

}
```


Primele câteva linii de cod din Listingul 25.6 sunt acestea:

```
$HOME = (getpwnam("www03640"))[7];
$path = "$HOME/survey";
$database_file = ">>$path/unKNOWn.dat";
$quote = "\"";
$ASCII_ten = pack("c", 10);
$ASCII_thirteen = pack("c", 13);
$space = " ";
```

Acest bloc atribuie valori unor variabile pentru a fi folosite ulterior în program.

În următorul grup de instrucțiuni, sunt apelate subrutinele dezvoltate în secțiunea precedentă. Prima linie apelează subrutina `User_Data` și stochează datele returnate în tabloul asociativ `%data_received`:

```
%data_received = &User_Data
```

Apoi, este apelată subrutina `No_SSI`, tabloul asociativ `%data_received` fiind transmis subrutinei.

După ce au fost apelate aceste două subrutine, datele din formularul utilizatorului au fost primite și decodificate. Scriptul CGI poate, acum, să se concentreze asupra celui de al doilea task, anume să adauge datele utilizatorului în baza de date de tip fișier text.

Primul pas din acest proces este să efectuați orice formatare specială care s-ar putea face asupra datelor. În acest caz, singura formatare necesară este eliminarea oricăror caractere carriage return (retur de car) sau newline (linie nouă) pe care utilizatorul le-ar fi putut introduce. Această formatare se efectuează cu următoarele linii de cod Perl:

```
foreach $key (keys(%data_received)) {
  $data_received{$key} =~ s/$ASCII_ten//ge;
  $data_received{$key} =~ s/$ASCII_thirteen/$space/ge;
}
```

Ele baleiază fiecare element din tabloul asociativ `%data_received`, căutând și eliminând caracterele carriage return sau newline.

Următorul pas este să deschidem fișierul bază de date, ceea ce se face cu următoarea instrucțiune Perl:

```
open(DATABASE,"$database_file") ||
  die "Content-type: text/html\n\nCannot open database.";
```

Deoarece serverul Web ar putea primi mai multe cereri pentru un script CGI în același timp, s-ar putea ca multe copii ale scriptului `survey.pl` să ruleze în oricare moment. De aceea, este nevoie să vă asigurați că o singură instanță a scriptului dumneavoastră CGI modifică fișierul bază de date la un moment dat. Pentru a preveni ca alte instanțe ale fișierului `survey.pl` să modifice fișierul bază de date, ar trebui să plasați un zăvor pe fișier imediat după deschiderea sa. Aceasta se efectuează cu instrucțiunea Perl `flock`:

```
flock(DATABASE, 2);
```

Cel de al doilea parametru al funcției `flock`, în acest caz 2, este un cod care determină tipul de zăvor care urmează să fie plasat peste fișier. Numărul 2 desemnează stabilirea unui zăvor la scriere pentru fluxul fișierului specificat.

Cu fișierul bază de date deschis și blocat, următorul pas este să scrieți datele utilizatorului în fișierul bază de date. Acest pas se efectuează cu instrucțiunea `print` din Listingul 25.6, elementele din tabloul asociativ `%data_received` fiind printate unul o dată, încadrate de paranteze și separate de virgule.

Singurul pas care mai rămâne după plasarea înregistrării în fișierul bază de date este să eliminați zăvorul de pe fișier și să închideți fluxul fișierului, ceea ce se face cu următoarele linii:

```
flock(DATABASE, 8);
close(DATABASE);
```

Eliminarea zăvorului implică un alt apel la funcția `flock`, de această dată cu al doilea parametru stabilit la 8. Numărul 8 desemnează eliminarea tuturor zăvoarelor din fluxul de fișier specificat. Fluxul de fișier este, apoi, închis folosind instrucțiunea Perl `close`.

Ca în cazul tuturor aplicațiilor CGI, fișierul `survey.pl` are nevoie să returneze un rezultat browserului Web al utilizatorului. Pentru această situație, cel mai ușor mod de a returna un rezultat este să redirectați browserul Web al utilizatorului către o pagină Web existentă, care conține un mesaj ce permite utilizatorului să știe că datele trimise au fost recepționate cu succes. Această redirectare este făcută trimițând directiva `Location` serverului Web, prin această instrucțiune Perl:

```
print "Location:http://www.ungroomd.com/newissue/admin/unknown-success.html\n\n";
```


Resurse pentru o informare sporită

Discuția despre CGI din acest capitol vă oferă o privire minimală asupra modului în care lucrează interfața CGI și în care o puteți implementa. Pentru alte informații asupra creării aplicațiilor CGI aveți la dispoziție mai multe surse. Majoritatea acestor resurse sunt disponibile, în versiune online, dar există și unele versiuni offline.

Resurse online

Când căutăm informații despre o tehnologie online, multiplele zone de informații de pe Internet sunt un loc minunat spre care să ne îndreptăm. Cu situri Web, grupuri de discuții și liste poștale, multe opțiuni vă sunt disponibile. Pentru informații de referință, cel mai bun loc de căutat este în siturile Web legate de subiect. Pentru răspunsuri la întrebări, totuși, adesea este mult mai util să apelăm la grupuri de discuții și liste poștale, pentru a obține răspunsuri rapide din partea altor utilizatori cunoscători.

Situri Web

Multe situri Web posedă informații despre crearea aplicațiilor CGI, inclusiv îndrumare (tutorials), manuale de referință și chiar arhive de scripturi. Ceea ce urmează reprezintă o listă a câtorva situri Web referitoare la CGI:

Overview of HTTP

`http://www.w3.org/hypertext/WWW/Protocols/`

Object Headers

`http://www.w3.org/hypertext/WWW/Protocols/HTTP/Object_Headers.html`

Names and Addresses, URIs, URLs, URNs, URCs

`http://www.w3.org/hypertext/WWW/Addressing/Addressing.html`

Uniform Resource Locators

`http://www.w3.org/hypertext/WWW/Addressing/URL/Overview.html`

HTML Quick Reference

`http://www.cc.ukans.edu/info/HTML_quick.html`

HyperText Markup Language

`http://www.w3.org/pub/WWW/MarkUp/`

A Beginner's Guide to HTML

`http://www.ncsa.uiuc.edu/General/Internet/WWW/HTMLPrimer.html`

Special Characters

`http://www.utoronto.ca/webdocs/HTMLdocs/NewHTML/entities.html`

ISO8859-1 (Latin-1) Table

`http://www.uni-passau.de/~ramsch/iso8859-1.html`

The Common Gateway Interface

`http://hoohoo.ncsa.uiuc.edu/cgi/`

W3C httpd CGI/1.1 Script Support

`http://www.w3.org/hypertext/WWW/Daemon/User/CGI/Overview.html`

Perl Home Page

`http://www.perl.com`

Perl Basics

`http://briet.berkeley.edu/perl/perl_tutorial.txt`

Perl for Win32

`http://www.perl.hip.com`

Index of Perl/HTML Archives

`http://www.seas.upenn.edu/~mengwong/perlhtml.html`

Boutell.com Archive Resources

`http://www.boutell.com/`

Matt's Script Archive

`http://www.worlwidemart.com/scripts/`

Selena Sol's CGI Script Archive

`http://www.extropia.com/`

C++ CGI Class Library

`http://www.ncsa.uiuc.edu/People/daman/cgi++/`

cgi-lib.pl Perl CGI Library

`http://www.bio.cam.ac.uk/cgi-lib/`

CGI.pm Perl 5 CGI Library

`http://www.genome.wi.mit.edu/ftp/pub/software/WWW/cgi_docs.html`

CGI Manual of Style

`http://www.mcp.com.zdpress/features/3970/`

WDB – A Web Interface to SQL Databases

`http://arch-http.hq.eso.org/bfrasmus/wdb/wdb.html`

Free Database Gateway List

`http://cuiwww.unige.ch/-scg/FreeDB/FreeDB.list.html`

WWW-DBMS Gateways

`http://grigg.chungnam.ac.kr/-uniweb/documents/www_dbms.html`

GSQL – A Mosaic-SQL Gateway

`http://www.ncsa.uiuc.edu/SDG/People/jason/pub/gsql/starthere.html`

Cu situri Web în mișcare sau care dispar, nu trebuie prea mult timp ca o listă statică precum aceasta să conțină unele legături expirate. De asemenea, apar mereu noi situri, oferind mai multe resurse și alte informații de ajutor. Dacă sunteți în căutare de informații care nu se găsesc pe unul din siturile listate aici, ar trebui să folosiți unul dintre cataloagele de căutare în Internet care urmează:

Yahoo!(`http://www.yahoo.com/`)

Infoseek(`http://www.infoseek.com/`)

AltaVista(`http://www.altavista.digital.com/`)

Excite(`http://www.excite.com/`)

Lycos(`http://www.lycos.com/`)

ClNet's Search.com(`http://www.search.com/`)

HotBot(`http://www.hotbot.com/`)

Grupuri de discuții și liste poștale

Grupurile de discuții și listele poștale reprezintă surse importante de informații asupra CGI și a tehnologiilor asociate. Deși siturile Web furnizează arhive și material de referință excelente, grupurile de discuții și listele poștale sunt mult mai bune pentru a căpăta răspuns la întrebări specifice. Puteți transmite întrebările dumneavoastră unui grup de discuții sau să trimiteți un mesaj e-mail către fiecare dintre cei care au subscris la o listă poștală și veți obține sfaturi, sugestii, uneori chiar cod concret ca răspuns la întrebarea dumneavoastră. Mai jos, sunt câteva dintre grupurile de discuții și listele poștale legate de CGI:

`comp.infosystems.www.authoring.cgi`

`comp.infosystems.www.authoring.html`

`comp.infosystems.www.authoring.misc`

`http://www.dejanews.com/` (Un utilitar de căutare prin grupurile de discuții Usenet)

Lista Common Gateway Interface

Pentru a subscrie, trimiteți un mesaj e-mail la `listserv@vm.ege.edu.tr`.

În corpul mesajului, tastați `subscribe CGI-L Firstname Lastname`.

Lista de discuții Advanced CGI

Pentru a subscrie, trimiteți un mesaj e-mail la `listproc@lists.nyu.edu`.

În corpul mesajului, tastați `subscribe ADV-CGI Firstname Lastname`.

Lista de discuții Advanced HTML

Pentru a subscrie, trimiteți un mesaj e-mail la `listserv@ua1vm.ua.edu`.

În corpul mesajului, tastați `subscribe ADV-HTML Firstname Lastname`.

Resurse offline

În plus față de bogăția de resurse online disponibile despre CGI și tehnologiile aferente, puteți găsi și multe materiale offline, din care multe sunt cărți tehnice despre subiect. Mai jos sunt câteva din cărțile mai populare despre subiectele CGI:

CGI Bible, IDG Books Worldwide

Teach Yourself CGI Programming with Perl 5 in a Week, Sams.net Publishing

Special Edition Using CGI, QUE

CGI Manual of Style, Ziff-Davis Press

The CGI Book, New Riders Publishing

Instant Web Scripts with CGI Perl, M & T Books

CGI Programming in C & Perl, Addison-Wesley Publishing Company

Adițional acestor titluri, sunt disponibile multe alte cărți despre CGI. Dacă nici una dintre ele nu se potrivește necesităților dumneavoastră, căutați la raionul cu cărți de calculatoare al librăriei dumneavoastră locale sau vizitați librăriile online `Amazon.com` (`http://www.amazon.com`) sau Barnes and Noble (`http://www.barnesandnoble.com`).

Rezumat

Common Gateway Interface, sau CGI, este un standard de aplicații pe care le dezvoltați pentru comunicația cu serverul dumneavoastră Web. Folosind această poartă de comunicație, puteți integra aplicațiile direct în situl dumneavoastră Web, extinzându-i astfel funcționalitatea. CGI nu este un limbaj de programare specific, așa că trebuie să alegeți unul cu care să vă scrieți aplicațiile CGI. Variantele obișnuite sunt Perl, C, C++, Visual Basic, Java, TCL, AppleScript și orice program shell UNIX.

O mare parte a CGI gravitează în jurul anteturilor trimise cu fiecare conexiune HTTP. Când un browser Web cere un document de la un server Web, el trimite o cerere HTTP către serverul Web. Această cerere conține un antet cu câmpuri de informații specifice. Când serverul Web răspunde, după aceea, cererii, de obicei prin returnarea documentului cerut, el trimite, de asemenea, un antet HTTP împreună cu răspunsul HTTP.

Cele mai multe date disponibile aplicațiilor dumneavoastră CGI provin din aceste anteturi de cerere HTTP. Datele tip formular trimise de browserul utilizatorului sunt, uneori, trimise aplicațiilor dumneavoastră CGI prin intrarea standard, când metoda cererii este metoda `POST`.

După ce aplicația CGI a dumneavoastră și-a completat taskul, aveți nevoie ca ea să transmită un răspuns înapoi către browserul Web. Pentru a face aceasta, un antet de răspuns HTTP trebuie să preceadă datele care sunt transmise browserului Web. Puteți adăuga antetul de răspuns într-unul din două moduri. Aplicația dumneavoastră CGI poate să creeze acest antet ea însăși și să trimită datele în mod direct înapoi către browserul Web. Sau aplicația CGI poate trimite datele și o directivă către serverul Web, care, apoi, formează antetul de răspuns HTTP și trimite toate informațiile înapoi serverului Web.

Întrucât CGI este o interfață de partea serverului, aplicațiile pe care le dezvoltați vor rula pe o mașină de partea serverului, ceea ce vă permite să beneficiați de avantajele creării de aplicații de partea serverului. Câteva dintre acestea sunt suportul deplin multiplatformă/multibrowser, flexibilitate sporită în taskurile pe care le realizați, putere de procesare crescută și integritatea codului sursă.

Din punct de vedere practic, ar trebui să utilizați CGI oricând aveți nevoie să integrați în situl dumneavoastră Web o aplicație pe care ați dezvoltat-o. Aplicațiile concrete pentru care folosiți CGI vor varia foarte mult, depinzând de anvergura sitului dumneavoastră Web și de ceea ce aveți nevoie să realizați. Unele aplicații comune folosesc aplicațiile CGI pentru a adăuga motoare de căutare, numărătoare de acces, rutine de tratare a formularelor și integrare de baze de date.

PARTEA a VI-a

HTML dinamic

Capitolul 26
În căutarea unui standard

de Rick Darnell

În acest capitol

- **Consorțiul World Wide Web 380**
- **Netscape 383**
- **Microsoft 385**

HTML dinamic – este ultimul strigăt al modei, dar știe în realitate cineva ce înseamnă? În toate conversațiile despre HTML dinamic (DHTML, Dynamic HTML), singurul lucru care lipsește este un standard identificabil, sau, mai precis, un standard cu care fiecare să fie de acord.

Aici este problema: Microsoft și Netscape au amândouă propriile lor idei despre cum ar trebui să arate DHTML, începând chiar cu numele. Microsoft numește implementarea sa Dynamic HTML (cu literă mare, *D*), iar Netscape se referă la implementarea sa ca dynamic HTML (cu literă mică, *d*). Istoria celor două companii conlucrând fericit la binele comun al tuturor utilizatorilor Web nu este parte a realității noastre.

În cazul DHTML, Microsoft și Netscape folosesc în mare parte o terminologie comună și se referă la același set de standarde propuse în munca lor de creare a paginilor Web interactive și dinamice. Ambele companii discută despre aptitudinea paginilor de a se modifica fără voiaje suplimentare către server și de un control mai extins al stilurilor. Dar aici este piatra de încercare: implementările actuale nu sunt total compatibile.

Netscape a adăugat o etichetă nouă, iar Microsoft a adăugat atribute noi pentru etichete existente. Netscape a adăugat tipuri de caractere cu încărcare din mers, iar Microsoft a adăugat controale ActiveX pentru accesul bazelor de date. Între cele două, apele devin mai mâloase, nu mai clare.

Vorbind despre standarde, este timpul să menționăm Consorțiul World Wide Web (W3C). W3C dezvoltă propria sa recomandare pentru DHTML. Totuși, conform cu realitatea spiritului de lucru în comitet, standardul W3C există de luni de zile ca o listă de posibilități, ceea ce înseamnă că toată munca depusă deocamdată de Microsoft și Netscape se bazează numai pe bunele lor presupuneri despre cum va arăta standardul final.

Deci unde vă situează aceasta? În același loc în care ați mai fost: încercați să dezvoltați pagini care sunt accesibile pentru cât de mulți oameni posibil, în timp ce lucrați cu implementări care, deși nu se exclud mutual, au puternice tendințe către incompatibilitate.

Pentru a vă ajuta să înțelegeți încotro este împinsă această întreagă tehnologie, iată o descriere a jucătorilor și a regulilor la care veți privi în acest capitol:

- W3C a dezvoltat o listă de posibilități pentru DHTML, la care se referă ca *Document Object Model* (Modelul de obiect document, DOM). Deși sună impresionant, DOM este, în realitate, doar o modalitate pentru browser de a identifica fiecare etichetă de marcare și element de pagină și de a le face accesibile pentru inspecție sau modificare. Mai multe informații despre planul DOM al W3C și despre implementările preliminare ale DOM în Microsoft și Netscape sunt tratate în Capitolul 27, „Modelul de obiect document".
- Netscape a creat propriul său model de obiect document acum câțiva ani prin Navigator 2.0. Mai recent, Netscape a încercat alte metode de a crea comportament dinamic, în primul rând prin utilizarea unei etichete `<LAYER>` nestandard. Totuși, Netscape a început să diminueze importanța noii etichete aproape de îndată ce a lansat versiunea finală de Netscape Communicator și lucrează acum în direcția unei implementări care să se dezvolte pe baza modelului său original de obiect document. Această abordare aduce Netscape mai aproape de W3C și Microsoft în privința acestui aspect al DHTML.
- Ca parte a ediției Internet Explorer 4.0, Microsoft a lansat versiunea sa pentru ceea ce crede că va fi W3C Document Object Model. Se pare că această versiune respectă lista de intenții a W3C, deși conține și unele caracteristici particulare, bazate pe controalele ActiveX.

OBSERVAȚIE

Abordarea divergentă este tipică pentru cele două principale companii producătoare de browsere, deoarece ele au încercat să anticipeze cum se vor dezvolta standardele DHTML. De exemplu, în 1996, atât Microsoft, cât și Netscape au lansat browserele lor respective în versiunea 3, care implementau standardul HTML 3.2, deși acesta nu a fost acceptat oficial decât în 1997.

Înainte de a trece la detaliile DHTML, în Capitolul 27, vom explora în acest capitol unele viziuni diferite asupra modului în care ar putea să arate DHTML când primul său standard oficial va fi aprobat de W3C.

Consorțiul World Wide Web

Deși nu seamănă a formă finală sau utilizabilă, ultimul cuvânt asupra funcționalității DHTML aparține Consorțiului World Wide Web și grupului său de lucru Document Object Model (DOM). Există un pic de ironie în această situație, deoarece unul dintre primele acte ale președintelui grupului a fost să emită o declarație care eticheta termenul *Dynamic HTML* drept „deocamdată marketing".

Grupul de lucru DOM a publicat o declarație de intenții, care afirmă că ținta sa nu este să extindă HTML sau să dezvolte un standard specific unui limbaj de scriptare. Conform paginii de bază pentru DOM de pe situl Web al W3C, grupul lucrează la o „interfață neutră față de platformă și limbaj, care permite programelor și scripturilor să acceseze dinamic și să actualizeze conținutul, structura și stilul documentelor".

Este ceva, dar ce înseamnă?

W3C a primit mai multe propuneri de la companii membre, în primul rând de la Microsoft și Netscape, despre felul în care diferitele elemente ale documentelor HTML ar trebui să fie așezate în scripturi. Este important de observat că nici una dintre propuneri nu are în vedere etichete HTML noi sau modificarea tehnologiei actuale a foilor de stil. Unul dintre cele mai importante și mai dificile scopuri ale W3C este să se asigure că nici o soluție nu favorizează tehnologia unei companii în defavoarea tuturor celorlalte.

Așa cum s-a menționat mai înainte în acest capitol, standardul W3C gravitează în jurul Modelului de obiect document, discutat în mai mare detaliu în Capitolul 27. În esență, modelul W3C are două cerințe de bază:

- Modelul de obiect document poate fi folosit pentru a lucra cu documentul, chiar după ce este încărcat de către browser. Elemente individuale și atributele lor pot fi adăugate, eliminate sau schimbate în cadrul documentului. Acest model include, de asemenea, un mod de a determina și schimba conținutul unei pagini, fie că reprezintă text, imagini, microaplicații sau programe plug-in, aceasta fiind partea dinamică despre care vorbesc toți.
- DOM nu va cere o interfață grafică de utilizator pentru implementare, datorită scopului W3C de a furniza standarde care să permită accesul la conținutul Web pentru toate tipurile de browsere, inclusiv acelea bazate pe tehnologiile Braille și audio. Amintiți-vă, DOM este un mod de deschidere a structurii și conținutului unei pagini Web către designerul de pagină astfel ca pagina să poată interacționa cu utilizatorul; nu este un standard pentru grafică sau animație, chiar dacă poate fi folosit în acest mod.

OBSERVAȚIE

W3C lucrează din greu să dezvolte standarde care nu exclud nici o platformă sau browser. Acest țel face parte din intenția de a oferi standarde World Wide Web care acceptă browsere specializate pentru persoane invalide, precum orbii și surzii. Din acest motiv, fiecare nouă etichetă sau mod de manipulare a unei etichete este dezvoltată cu un ochi către oferirea unui înlocuitor de tip text, care poate fi interpretat de browsere pentru nevoi speciale.

Pentru a oferi suport acestui efort, W3C a lansat Inițiativa pentru accesibilitatea Web (Web Accesibility Initiative, WAI) în vederea promovării și a realizării funcționalității Web pentru persoane cu infirmități. Inițiativa implică înființarea unui Birou de programe internaționale (International Program Office, IPO), care, conform unui comunicat de presă al W3C, este responsabil cu „dezvoltarea tehnologiilor și protocoalelor software, creând linii directoare pentru utilizarea tehnologiilor, formarea conduitei industriei și dirijarea cercetării-dezvoltării". Întrucât biroul IPO este parte a W3C, el va asigura și ca toate noile standarde și tehnologii W3C să îndeplinească sau să depășească scopurile de accesibilitate.

Pentru mai multe informații despre HTML și accesibilitate, vezi Capitolul 41.

Toate celelalte cerințe ale DOM provin din liniile directoare asupra accesibilității și manipulării, incluzând fiecare zonă importantă reclamată de grupul de lucru DOM:

- **Navigarea prin structură**: Aceasta reprezintă posibilitatea de a repera elemente într-un document, cum ar fi părinții sau copiii unui obiect, și este modul în care Netscape a pornit modelul său de document în Navigator 2.0. Se începe cu `window`, se trece în jos la `document`, apoi la diverșii copii de `document`, incluzând `form`, `link`, `applet` și alte elemente de pagină. Folosind modelul DOM, sunt expuse toate etichetele pentru browser, inclusiv etichete și elemente necunoscute.
- **Manipularea documentului**: Standardul prevede un mod de a adăuga, a elimina sau a modifica elemente și etichete, inclusiv atribute de etichete, în interiorul documentului.
- **Manipularea conținutului**: Aceasta reprezintă posibilitatea de a adăuga, de a modifica sau de a șterge conținut din document sau etichete individuale. Include și o cerință pentru a determina, din oricare parte a documentului, care etichetă afectează textul.
- **Modelul evenimentului**: Modelul evenimentului este suficient de intuitiv pentru a genera documente complet interactive. El include capacitatea de a răspunde oricărei acțiuni a utilizatorului în interiorul documentului, incluzând mișcarea în sau afară din câmpurile de formular, detectarea mișcării și a clicurilor mouse-ului, determinarea oricărei apăsări de taste. Cu toate că W3C s-a angajat la realizarea accesibilității pentru persoane cu infirmități, unele dintre evenimente se vor aplica numai unei interfețe grafice (cum ar fi Windows), care este destinată utilizatorului obișnuit.

■ **Modelul obiectului foaie de stil**: Acesta este asemănător manipulării documentului, menționată mai înainte. Sub DOM, atributele foii de stil în cascadă sunt, de asemenea, expuse pentru modificare. Cu un ochi spre viitor, W3C include o rezervă pentru a extinde modelul foii de stil către alte formate. Aceasta ar putea fi portița de care Netscape are nevoie pentru a include foile de stil JavaScript în timp ce menține conformitatea cu standardele W3C.
■ **Informații generale despre document și browser**: W3C nu a lăsat nici o piatră neîntoarsă. Parte din DOM include capacitatea de a examina obiecte înglobate, cum ar fi prăjituri și data la care un document a fost creat. Alte informații disponibile includ marca și versiunea agentului de utilizator (browserului) și tipurile MIME pe care le suportă.

Setul complet de cerințe pentru DOM acoperă circa trei pagini și include toate articolele precedente, plus definiții de tip și raportarea erorilor. În esență, toate cerințele se reduc la aceasta: *totul* într-un document trebuie să fie disponibil interfeței de utilizator pentru manipulare.

Primul pas către crearea standardului este să se estimeze care este modelul de obiect curent. Acest pas folosește modelele implementate de Internet Explorer 3.0 și Navigator 3.0, care au dus la standardul de nivel 0, neprezentat publicului. A fost folosit numai ca referință internă de către membrii comitetului.

După ce a fost creat standardul de nivel 0, grupul de lucru a început munca pentru atingerea țelului său pe termen lung: stabilirea unui consens asupra a ceea ce ar trebui și ce nu ar trebui să fie parte a standardului Modelul de obiect document, care va deveni nivelul 1. Nivelul 1 include toate caracteristicile de bază de manipulare a structurii și conținutului. Alte caracteristici, ca manipularea DTD și un set extins de evenimente, vor fi parte a versiunii de nivel 2.

Cât vor dura toată această scărpinare în cap și pasare a hârtiilor? Președintele grupului de lucru, Lauren Wood, afirmase că o versiune finală pentru DOM Level 1 ar putea fi lansată la data tipăririi acestei cărți (decembrie 1997), iar nivelul 2 ar putea apărea până la sfârșitul lui 1998 sau începutul lui 1999.

Această discuție ne duce înapoi la întrebarea de început: ce are de făcut programatorul? Răspunsul din partea W3C este să vă scrieți paginile DHTML la cel mai scăzut numitor comun de compatibilitate până ce consensul va începe să se formeze. Acest sfat este greu de înghițit pentru dezvoltatori, în special când doriți să scrieți cu instrumentele „cele mai moderne și mai importante" din Web.

Recomandarea mea este similară celei de la W3C: dacă scrieți pentru o audiență generală, fiți prudent cu orice este specific, incluzând etichete sau atribute care sunt domeniul exclusiv al unui browser sau al altuia. Veți învăța mai multe despre aceasta mai târziu, dar pentru început veți folosi, probabil, o combinație de foi de stil și JavaScript. Pentru a vă asigura că implementarea dumneavoastră este robustă, testați-o cu o largă varietate de browsere: versiuni noi și vechi de Navigator și Internet Explorer, versiuni fără zorzoane, ca Mosaic, și aplicații numai pentru text, precum Lynx. Această măsură de prevedere este necesară pentru a vă asigura că paginile dumneavoastră dinamice nu vor forma un crater în browserul nondinamic al cine știe cui.

Dacă doriți să scrieți la nivelul standardului în curs de elaborare și cu cele mai avansate instrumente disponibile, folosiți o zonă separată pe situl dumneavoastră Web cu marca „for demonstration only" (numai pentru demonstrație), împreună cu o notă despre browserul pentru care a fost scrisă.

În concluzie, mai este un drum lung până la o implementare care să funcționeze bine cu toate browserele, în special cu propunerile majore de la Netscape și Microsoft. Dacă scrieți la capacitățile majorității utilizatorilor dumneavoastră și includeți plase de siguranță pentru restul, nu veți greși.

Netscape

Mai demult (cam prin 1995) Netscape a semnat actul de naștere al obiectului DHTML prin lansarea ediției Navigator 2.0. Respectiva versiune includea o caracteristică numită JavaScript, care oferea un model de obiect document care dădea acces la elemente ca formulare, hiperlegături, culori și diferite atribute de browser.

Primul model de obiect a fost creat cu o ierarhie de instanțe, care reflecta construcția paginii HTML. La vârful ierarhiei era `window`, părintele tuturor celorlalte obiecte. Acest obiect avea patru copii: `location`, `history`, `frames` și `document`. Obiectul `document` mai includea și copii reprezentând informații specifice despre document:

```
alinkColor
bgColor
fgColor
linkColor
vlinkColor
cookie
lastModified
location
anchors
referreru
forms
links
title
```

O ierarhie de instanțe este construită din instanțe concrete, mai degrabă decât din clase generale, de obiecte. De exemplu, să presupunem că singurele elemente permise pe o pagină sunt titluri (headings) și o anume pagină are trei H1 și un H4. O ierarhie de clase ar poseda un obiect pentru fiecare antet posibil, de la H1 la H6, chiar și pentru cele nefolosite, H2, H3, H5 și H6. Într-o ierarhie de instanțe, aceeași pagină ar include un obiect doar pentru titlurile care apar efectiv în document.

După cum puteți vedea, implementarea DOM de la Netscape era un amestec de atribute de pagină (configurații de culoare), informații META (data ultimei modificări și pagina de referință) și elemente fizice (ancore, formulare și legături). În versiunea următoare de Navigator (3.0), modelul de obiect a fost extins pentru a include miniaplicații și alte obiecte înglobate.

Netscape a avut o idee bună, dar care a fost puțin limitată în scop și utilizare. Numai câteva dintre elemente, precum elementele de formular și unele de culoare, puteau fi modificate fără reîncărcarea documentului. Era, de asemenea, posibil să modificați restul cuprinsului unei pagini, dar numai prin reîncărcare cu protocolul `javascript`, care este folosit în același mod ca alte protocoale Internet, cum ar fi `http` și `ftp`. Utilizatorul încă vede că pagina apare

albă și apoi se reîncarcă. Totuși, este mult mai rapid decât refacerea unei versiuni noi de la server, deoarece pagina este creată de un proces din browser.

Vorbind de protocol, ar fi o dovadă de neglijență să nu menționez JavaScript drept cealaltă jumătate a marii realizări de la Netscape cu Navigator 2.0. Folosind JavaScript, dezvoltatorii pot scrie mici aplicații care rulează în browserul utilizatorului, în locul procesării prin intermediul serverului. Sintaxa a fost preluată din Java, iar o dată cu Netscape 3.0 cele două limbaje au putut comunica înainte și înapoi de-a lungul paginii Web. Această realizare a fost un mare succes, deoarece a oferit pentru Java acces direct la datele de pe o pagină Web și a furnizat un mod de a controla codul Java din exteriorul unei miniaplicații.

La început, programatorii au folosit limbajul JavaScript și modelul de obiect document Netscape în special ca să verifice conținutul formularelor sau să facă mici calculatoare 1040EZ pentru distracție. Câțiva temerari l-au folosit să creeze unele efecte elegante, precum expandarea și contractarea contururilor sau situri Web cu controale personalizate, dar cei mai mulți dintre utilizatori erau limitați în intenție și utilitate.

Lansarea ediției Communicator de către Netscape nu a extins modelul de obiect document în nici o direcție nouă și dramatică, cum au făcut cele anterioare, dar aceasta nu înseamnă că Netscape nu este în continuare cu ochii pe DHTML. Iată cele trei componente ale viziunii Netscape asupra DHTML:

- Folosirea layerelor pentru a muta, ascunde sau arăta blocuri de HTML pe pagina Web. Layerele pot fi manipulate ca răspuns la evenimentele de utilizator, transformându-le în singura porțiune din soluția Netscape care îndeplinește definiția de *dinamic*, adoptată mai înainte în acest capitol.

Un *layer* Netscape este o porțiune de cod HTML care este afișat, ascuns, mutat și modificat în diferite moduri. În esență, el convertește documentul dumneavoastră HTML într-un set de imagini tip diapozitiv pe care le puteți amesteca și afișa în orice ordine, individual sau în combinație.

- Control exact asupra formatării, fonturilor, machetării și altor aspecte ale comportamentului paginii prin intermediul foilor de stil. Netscape include suport pentru specificația W3C CSS1 (Cascading Style Sheets – Level 1, Foi de stil în cascadă – nivelul 1), pe care o folosește Microsoft, și JavaScript Style Sheets (JSSS, Foi de stil JavaScript). JSSS nu permite schimbarea documentului după ce a fost încărcat, dar permite unei foi de stil să chestioneze browserul despre mediul său. Foaia de stil poate, apoi, să aducă modificări implementării sale pentru a se potrivi situației specifice.

OBSERVAȚIE

Chiar dacă JSSS nu este acceptată în mod explicit de W3C, parte din poziția W3C asupra Modelului de obiect document este să extindă modelul foii de stil CSS1 asupra altor formate de foi de stil.

- Fonturi dinamice, care sunt atașate unei pagini Web în loc de a fi dependente de opțiunile disponibile pe calculatorul unui utilizator. În trecut, era necesar ca programatorii să ghicească ce tipuri de caractere erau disponibile pe mașina unui utilizator, iar apoi să furnizeze o listă de preferințe pentru o pagină cu ajutorul atributului `font` sau să folosească eticheta `<FONT>`. Caracteristica de fonturi dinamice utilizează o metodă nouă de a verifica existența tipului de caractere dorit și, dacă acesta nu este găsit, îl descarcă de pe server.

W3C lucrează la un standard de fonturi dinamice care va permite browserelor Web să descarce rapid o copie a fonturilor folosite într-o pagină Web. Această caracteristică ar

elimina unele dintre problemele cu disponibilitatea fonturilor, care au devenit din ce în ce mai vizibile cu foile de stil.

Ar mai fi ceva de spus despre layere. În versiunile inițiale beta de Navigator 4.0, Netscape a decis să utilizeze o etichetă nouă, `<LAYER>`, pentru a implementa poziționarea precisă a elementelor. Această decizie a cauzat o oarecare problemă întrucât Microsoft nu intenționa să includă eticheta în browserul său, astfel că W3C a decis să nu dezvolte eticheta `<LAYER>` pentru a funcționa cu alte abordări ale DHTML. Dată fiind o primire nu prea călduroasă, Netscape pune mai mult accent pe layere implementate prin Modelul de obiect document cu eticheta `<DIV>`. De asemenea, Netscape nu mai acordă atâta importanță etichetei `<LAYER>`, chiar dacă aceasta mai este parte a versiunii Netscape Communicator.

Soarta etichetei `<LAYER>` este neclară în acest moment. W3C nu dă vreun semn că va include suport pentru ea în specificația propusă pentru HTML 4.0, chiar dacă eticheta pare că îndeplinește multe dintre caracteristicile conceptului W3C asupra elementelor HTML de poziționare prin folosirea foilor de stil (`www.w3.org/pub/WWW/TR/WD-positioning`), la elaborarea cărora Netscape a pus umărul împreună cu Microsoft.

În DHTML, Netscape pare să coopereze cu W3C, în loc de a încerca să stabilească propriile sale standarde și să facă consorțiul W3C să le urmeze. Netscape Communicator este compatibil întrutotul cu recomandările W3C existente sau standardele pentru HTML 3.2, plus unele din HTML 4, inclusiv CSS. Ea încearcă și să mențină compatibilitatea cu conceptele în lucru asupra poziționării, modelelor de obiect și fonturilor dinamice.

În afara implementării foilor de stil JavaScript, tratată în Capitolul 21, „Folosirea foilor de stil JavaScript", și etichetei `<LAYER>`, tratată în capitolul 22, „Layere", Netscape pare să lucreze la o versiune de DHTML care nu depinde exclusiv de controale patentate sau specifice platformei.

Înseamnă aceasta că o pagină DHTML care lucrează cu Netscape va lucra și cu Microsoft dacă ambele browsere susțin că îndeplinesc standardele? Nu în mod necesar. Există mai multe puncte în litigiu aici. În primul rând, W3C nu leagă DHTML de nici un limbaj specific de scriptare. De exemplu, să presupunem că Microsoft face VBScript și Netscape face JavaScript să se înțeleagă complet cu standardul DHTML. Dacă un browser Netscape încarcă o pagină realizată dinamic cu VBScript, nu este de așteptat să funcționeze, deoarece Navigator încă nu suportă acest limbaj specific.

În al doilea rând, ambele companii văd standardul în moduri diferite și pretind privilegiul exclusiv de a fi singura „soluție adevărată". Această diferență a punctelor de vedere reprezintă motivul pentru care nu există un model de obiect document extins de o parte și layere de cealaltă parte. Aceste soluții nu sunt compatibile cu diferitele browsere, chiar dacă ele, amândouă, întrunesc litera sau spiritul muncii W3C. În mod curent, nici un browser nu a fost scris pentru a suporta vederile celuilalt asupra subiectului, chiar dacă fiecare pretinde a fi „flexibil cu standardul".

Microsoft

Deoarece a devenit practica de operare standard în continua bătălie a browserelor, Microsoft se îndreaptă cu îndrăzneală acolo unde nici un standard nu a ajuns înainte cu definirea și implementarea DHTML în Internet Explorer 4.0. Puteți citi totul despre viziunea Microsoft asupra DHTML la `www.microsoft.com/workshop/author/dhtml/`.

Internet Explorer 4.0 (IE4) nu este doar o colecție de suporturi pentru câteva etichete noi și o interfață de utilizator nouă. Microsoft a revizuit complet motorul de analiză (parsing

engine) HTML. Motorul de analiză într-un browser Web se uită în documentele HTML să vadă ce este în interiorul lor. El se uită la textul etichetat și la conținut, iar apoi formatează conținutul bazându-se pe ceea ce găsește în etichete. Cu cuvinte simple, un analizor sparge unități mari de informație (cum ar fi o pagină Web) în bucăți mai mici (ca etichetele de marcare), care sunt mai ușor de interpretat și procesat, în același fel în care dumneavoastră citiți o carte.

Motorul de analiză extins IE4 ia etichetele individuale și elementele de document și le integrează cu un model de obiect document care este acceptat de oricare limbaj de scriptare disponibil. Din punct de vedere practic, cele două variante sunt JavaScript și VBScript, care sunt prevăzute drept componente ale IE4. Exemplele din această carte sunt create cu JavaScript, atât pentru Netscape, cât și pentru Microsoft.

SFAT

IE4 este o aplicație bazată pe componente care permite adăugarea altor limbaje de scriptare prin folosirea de programe terțe legate de browser. Deci, dacă puteți găsi module suplimentare, puteți să adăugați limbaje ca Python browserului dumneavoastră și să extindeți opțiunile dumneavoastră de scriptare.

Viziunea Microsoft asupra DHTML include patru componente, care sunt asemănătoare celor de la Netscape:

- Modelul de obiect document permite ca oricare element de pe pagină să fie arătat, ascuns, schimbat sau rearanjat fără a reîncărca pagina de la un server. Una dintre surprizele aduse de implementarea Microsoft a modelului DOM o reprezintă faptul că nu este specifică pentru Windows sau ActiveX. Este ca și cum modelul Microsoft ar fi complet înglobat în analizorul HTML ca parte a browserului. Prin implementarea DOM cu oricare limbaj de scriptare în browser, programatorii pot folosi limbajele de scriptare spe a examina sau controla practic orice element de pagină.
- Un mod de a controla elementele pe o pagină prin scripturi, cum ar fi JavaScript și VBScript. Limbajele de scriptare includ obiecte care se referă la elementele de pe pagină așa cum sunt definite de modelul de obiect document. Folosind caracteristicile de comparare și atribuire ale limbajului, dezvoltatorii de pagină pot examina și schimba atributul sau conținutul oricărui element de pe pagină specificat de DOM, cum ar fi titluri, legături, text și alte articole. Această capacitate include de asemenea controlul și manipularea miniaplicațiilor Java înglobate, controale ActiveX și programe plug-in.
- Controale multimedia ActiveX pentru animație și alte efecte, precum filtre și tranziții, fără a se bizui pe descărcarea de fișiere sau pagini mari. Aceste controale au fost implementate prin foi de stil în cascadă cu folosirea filtrelor fade-in, fade-out și a altor tipuri de efecte.
- Un mod de legare a datelor de o pagină HTML, inclusiv generarea automată de tabele, sortarea datelor tabulare și interogarea tabelelor locale. Această componentă folosește un set special de atribute HTML activate cu ActiveX pentru a afișa „pe viu" înregistrări din baza de date în același fel în care ați afișa o microaplicație sau un program plug-in. În vechea școală HTML, afișarea înregistrărilor bazei de date cerea lucrul cu scripturi CGI și programare avansată în Perl, C ori Java. Deoarece DHTML suportă modificări imediate în conținutul paginii, baza de date se poate actualiza automat ca răspuns la tastări ale utilizatorului sau tabelul poate afișa actualizări pe viu conform modificărilor făcute bazei de date de terți.

Ca și Netscape, Microsoft lucrează cu standarde și propuneri de standarde W3C, încercând să anticipeze viitorul DHTML-ului în lucrările consorțiului.

DOM de la Microsoft este foarte asemănător modelului Netscape Navigator 2.0. El începe cu obiecte principale reprezentând pagina și browserul (cum ar fi `window` și `document`), iar apoi se împarte în mai multe detalii (cum ar fi `form`, `applet` și `link`). Microsoft a extins în continuare modelul spre a include fiecare element de pe pagină, de la titluri la paragrafe, imagini, tabele sau linii orizontale de separare – accesul este neîngrădit.

Una dintre marile forțe trâmbițate de Microsoft este compatibilitatea cu browserele care nu acceptă implementarea DHTML din Internet Explorer 4.0. Microsoft numește compatibilitatea sa cu alte browsere *degradare elegantă*. Dezvoltatorii și autorii care scriu special pentru versiunea Microsoft de DHTML nu vor avea nevoie să producă versiuni alternative pentru Netscape sau altă firmă, conform Microsoft (vezi secțiunea „Realitatea degradării elegante"). Scripturile și etichetele folosite de Microsoft sunt tehnologii acceptate în uzul altor browsere și binecuvântate de W3C.

REALITATEA DEGRADĂRII ELEGANTE

Nu există pădure fără uscături, și DHTML nu face excepție. DHTML de la Microsoft nu se degradează elegant când este plasat pe Navigator 4.0, în special când depinde de foi de stil. Priviți Figurile 26.1 până la 26.3. În Figura 26.1, vedeți o panoramă DHTML din situl Microsoft așa cum se vede în Internet Explorer 4.0. Executați clic pe un titlu, și apare conținutul de dedesubt.

Primul test al degradării elegante este bătrânul de încredere Mosaic de la NCSA. Aceeași pagină încărcată în el (vezi Figura 26.2) se degradează destul de bine. Mosaic nu înțelege foile de stil sau scriptarea, așa că ignoră toate etichetele și accesoriile suplimentare și afișează pagina într-o manieră curată, directă, predictibilă pentru a face totul accesibil utilizatorului.

În Figura 26.3, vedeți aceeași pagină pe Navigator 4.0 (Preview Release 4). Microsoft utilizează atributul `display:none` pentru a ascunde conținutul până când utilizatorul execută clic pe titlu. Când se întâmplă acest lucru, o metodă JavaScript șterge atributul, astfel încât conținutul să poată fi afișat. Aici este problema: Netscape nu suportă schimbarea foii de stil CSS1 cu JavaScript, deci nu există nici o modalitate de a face să apară elementele ascunse fără încărcarea unei noi pagini de la server.

De ce apare lista marcată, chiar dacă este asociată cu conținutul de dedesubtul titlului? Deoarece Navigator nu crede că ea aparține de restul textului ascuns și înțelege că atributul de afișare `<UL>` este stabilit pentru a afișa. Rezultă o listă fără vreo explicație, ceea ce nu mai este o interpretare acceptabilă a paginii.

FIGURA 26.1.
Executați clic pe un titlu din pagina Microsoft DHTML și conținutul de mai jos apare.

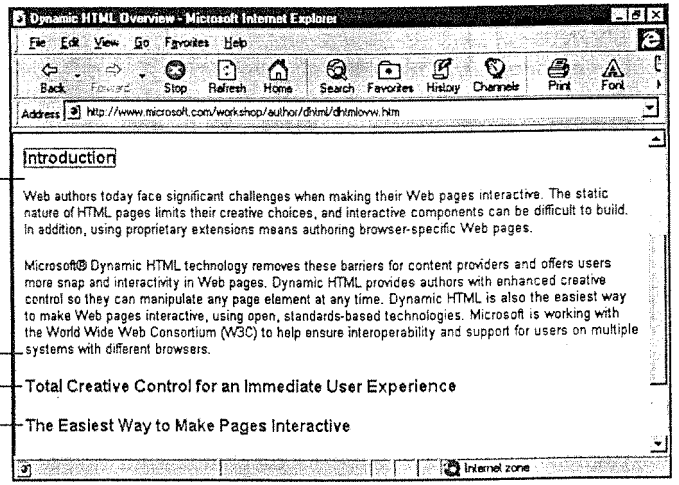


FIGURA 26.2. *Degradare elegantă pe un browser care nu este compatibil cu foile de stil CSS1 sau JavaScript.*

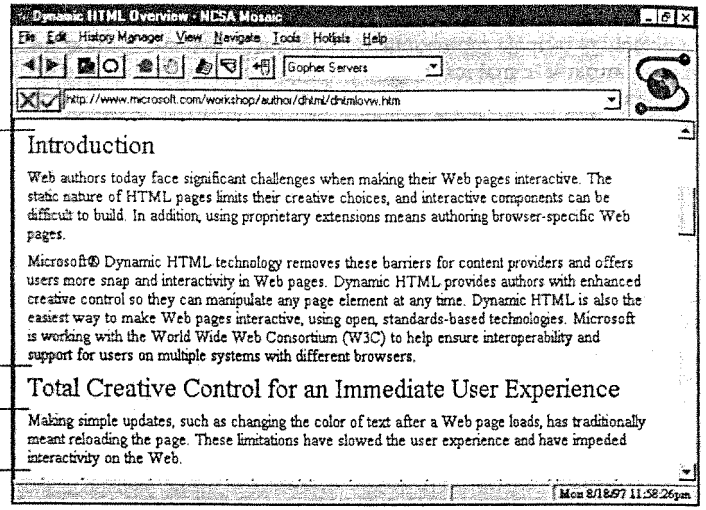
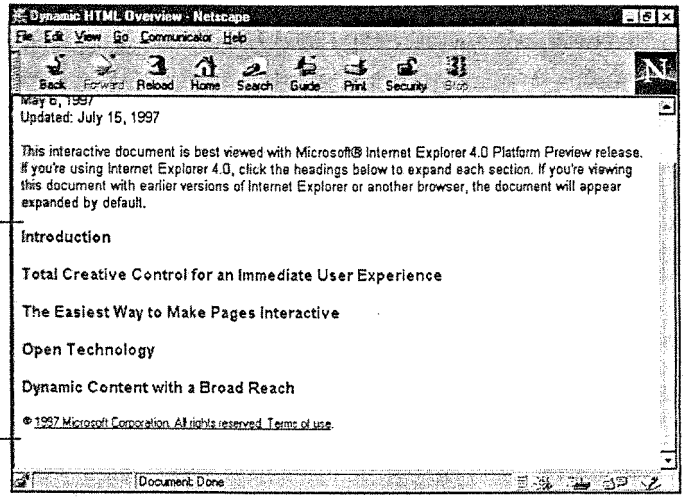


FIGURA 26.3. *Deoarece Netscape este compatibil cu CSS1, dar nu cu modificarea CSS1 de către JavaScript, pagina nu se degradează elegant.*



Rezumat

Un standard HTML dinamic ar trebui să rămână o țintă mobilă, deoarece producătorii de software continuă să revizuiască produsele lor pentru a îndeplini regulile schițate de W3C. În prezent, există trei școli de gândire ușor diferite asupra a ceea ce este DHTML.

Cei trei jucători principali – Consorțiul World Wide Web, Netscape și Microsoft – par să agreeze faptul că un model extins de obiect document este centrul soluției. Modelul extins de document va permite acces la fiecare element, împreună cu atributele și conținutul său, prin intermediul unui limbaj de scriptare. Cu acces neîngrădit și control asupra conținutului, paginile HTML vor părea făcute din plastilină, nu din piatră.

Plecând la DOM, vederile devin divergente. W3C nu folosește DHTML ca parte a vocabularului său deloc, cu excepția recunoașterii folosirii termenului de către vânzătorii de software. Munca sa este centrată aproape exclusiv pe dezvoltarea Modelului de obiect document (DOM) și pe legătura sa cu foile de stil, limbajele de scriptare și alte tehnologii existente.

Netscape include foile de stil JavaScript și layerele în pachetul său de posibilități DHTML. Se pare că layerele și JSSS sunt rezultatul unei implementări a schiței de lucru elaborate de W3C pentru poziționarea elementelor HTML, măcar că eticheta <LAYER> nu este inclusă ca parte a propunerii de specificație HTML 4.0.

În plus față de Modelul de obiect document, Microsoft pune accentul, cu viziunea sa de DHTML, și pe efectele multimedia și conștientizarea datelor. Totuși, soluția Microsoft depinde, și ea, de elemente patentate, precum controale ActiveX, pentru a implementa acces la DOM.

Spre deosebire de situația de acum unul sau doi ani, când cele două companii căutau să construiască standarde Web în jurul viziunilor proprii, atât Netscape, cât și Microsoft par mult mai dornice de a lucra cu Consorțiul World Wide Web în dezvoltarea standardelor pentru browserele lor. Se speră că au trecut zilele când fiecare companie dorea să creeze propriile sale etichete și tehnologii într-o încercare de a forța standardele să se conformeze respectivului său software. Deși competiția dintre cele două rămâne aprigă, utilizatorul nu mai este supus neplăcerilor create de incompatibilități.

Pare ca și cum Netscape și Microsoft ar putea să devină mai apropiate decât le-ar plăcea să admită. Pentru timpul prezent, ar fi bine să alegeți cel mai mic numitor comun al celor două – o combinație a Modelului de Obiect Document W3C, foi de stil în cascadă și scriptare – și să vă agățați de el până când recomandarea W3C va fi finalizată și diferențele dintre browsere se vor restrânge. Este încă prea devreme în acest joc să vă bazați pe elementele cu prea multe caracteristici de proprietar.

Până una alta, voi face o presupunere îndrăzneață și voi spune că recomandarea finală W3C va fi mai apropiată de Microsoft decât de Netscape, iar zonele comune dintre cele două nu vor fi complet compatibile. În acest punct, ambele companii vor avea nevoie să se asigure că, oricare ar fi implementarea lor DHTML, este în deplină concordanță cu standardul. Totuși, W3C se mișcă adesea mult mai lent decât restul lumii World Wide Web, astfel încât Microsoft și Netscape vor avea probabil destul timp pentru a se adapta standardului W3C, oricum ar arăta el.

Capitolul 27
Modelul de obiect document

de Rick Darnell

În acest capitol

- **Puțină istorie DOM 390**
- **În căutarea unui standard revizuit 394**

Modelul de obiect document (DOM, Document Object Model) sună destul de intimidant, sugerând camere pline de fum în care programatorii își freacă cu satisfacție mâinile și bolborosesc incantații deasupra cazanelor cu procesoare 8088 topite. Nu, stați o clipă; acelea erau vrăjitoarele din Macbeth.

În acest capitol, este treaba mea să risipesc credințele voodoo care înconjoară DOM și să îl prezint drept ceea ce este: o modalitate structurată de accesare a elementelor HTML și a conținutului unei pagini Web. Nu există nici un set nou de etichete sau atribute de învățat, deoarece ținta DOM este să funcționeze în interiorul HTML, al foilor de stil și al limbajelor de scriptare existente.

În acest capitol, veți arunca o privire spre istoria DOM cu Netscape și Microsoft, veți afla ce prevede Consorțiul World Wide Web (W3C) pentru viitor și veți vedea cu ce urmează să lucrați în restul cărții.

Puțină istorie DOM

DOM nu este un concept nou. El a apărut încă din 1995, o dată cu Netscape Navigator 2.0, care conținea una dintre primele sale implementări. În afară de aceasta, versiunea Netscape nu a strânit un interes prea mare decât în legătură cu folosirea limbajului JavaScript pentru formulare. O implementare nouă a DOM în Microsoft Internet Explorer 4.0 extinde modelul creat inițial pentru Netscape.

Deșteleniрea pământului – Netscape

Netscape a fost prima companie care a dezvoltat un model DOM prin crearea limbajului său de scriptare JavaScript. Ca primă încercare, modelul oferea multă funcționalitate pentru programatori. Inițial, DOM a fost descris ca o *ierarhie de instanțe* a *obiectelor JavaScript*, termenii fiind definiți mai jos:

- Instanță: Nici un obiect nu este creat până ce nu apare pe o pagină Web. De exemplu, dacă n-ar fi nici o hiperlegătură pe o pagină, obiectul `link` ar fi gol. Ierarhia de instanțe este diferită de o ierarhie generală de clase, în care un obiect este creat pentru fiecare clasă, fie că este inclus ca parte a paginii Web, fie că nu.
- Ierarhie: Nu toate obiectele sunt create egale. Fiecare există într-o relație stabilită cu alte obiecte (vezi Figura 27.1). În cazul unui browser Web, cel mai înalt loc în lanțul ierarhic este obiectul `window`. Sub `window` este un obiect `document`, care este pagina HTML curentă încărcată în browser. În prezentarea sa inițială, există trei obiecte sub `document`: `anchor`, `link` și `form`. Sub obiectul `form` se găsesc elemente individuale de formular.
- JavaScript: JavaScript este un limbaj de scriptare pentru documente HTML. Scripturile sunt declanșate după ce utilizatorul manipulează ceva pe document, cum ar fi plasarea

mouse-ului peste o hiperlegătură, executarea de clic pe un buton sau schimbarea conținutului unui câmp de formular.
- Obiecte: Un obiect este o construcție cu proprietăți și metode asociate. Gândiți-vă la o casă: proprietățile sale ar putea include articole ca dormitoare, băi, etaje și metri pătrați. Primele trei proprietăți, ar putea, de asemenea, să fie obiecte cu propriile lor proprietăți, cum ar fi toalete, căzi, dușuri, podea și ferestre. Chiar dacă proprietățile descriu casa, metodele sunt cele care o modifică. De exemplu, o metodă de mărire a casei ar putea afecta celelalte proprietăți prin adăugarea de camere și suprafață. Puteți învăța mai multe despre JavaScript în Capitolul 23, „Scurtă privire asupra limbajului JavaScript", și Capitolul 21, „Folosirea foilor de stil JavaScript".

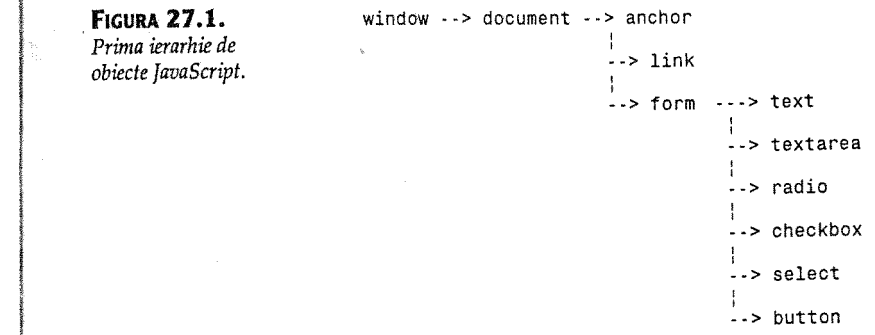


FIGURA 27.1. *Prima ierarhie de obiecte JavaScript.*

Acum, că ați văzut fiecare termen, puteți compune întreaga imagine. Mai întâi, DOM era o colecție de anumite elemente de pagină care existau într-o relație de descendență unul dintr-altul prin limbajul JavaScript. Cel mai de sus părinte în relație era fereastra browse-rului și ierarhia continua în jos până la elemente individuale pe pagină. Numai elementele selectate erau accesibile; erați în aer dacă doreați să lucrați cu etichete ca titluri sau citate bloc.

Netscape a decis să transforme această comportare nefirească într-un lucru bun la lansarea ediției Navigator 3.0. Principala caracteristică a noului JavaScript a fost existența obiectelor suplimentare reprezentate de programe plug-in, miniaplicații și imagini. Alte elemente de pe pagină, precum citatele bloc, rămâneau inaccesibile programatorilor JavaScript.

Prin introducerea lui Netscape Communicator cu Navigator 4.0, Netscape nu a extins JavaScript DOM în vreo manieră palpabilă. Puteți accesa formulare, miniaplicații, programe plug-in, imagini, layere, cadre, ancore și hiperlegături pe o pagină – componentele majore ale paginii – dar nimic altceva.

Navigator 4.0 include, de asemenea, suport pentru Foile de stil JavaScript (JSSS), care totuși folosesc un obiect `document` ușor diferit. JSSS folosește sintaxa JavaScript pentru a controla stilul unui document (citiți mai multe în Capitolul 21). Folosit în etichete `<STYLE>`, obiectul `document` adună un set întreg nou de obiecte, care nu sunt disponibile când este folosit în etichete `<SCRIPT>`. Noile obiecte se bazează pe o ierarhie de clase generală, în care este reprezentat fiecare element HTML, împreună cu clase de stil și identificatori singulari (vezi Figura 27.2).

Folosind această ierarhie de obiecte, puteți accesa și modifica oricare element HTML, fie că el apare pe pagină fie că nu. Ea permite dezvoltatorului de pagini să creeze funcții pentru a evalua browserul și mediul utilizatorului și a crea un stil bazat pe această cunoaștere.

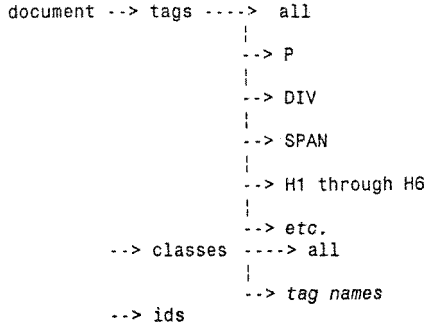


FIGURA 27.2. *Ierarhia de document JavaScript pentru foi de stil.*

Cu acestea în minte, e timpul să ne deplasăm pe Coasta de vest spre Washington și să vedem ce a realizat Microsoft cu DOM.

Extinderea ideii – Microsoft

Cu Internet Explorer 3.0, Microsoft a trecut de partea triumfătorului JavaScript, adăugând propriul său limbaj de scriptare (Visual Basic Scripting Edition, numit uzual VBScript) ca o altă opțiune. DOM, care era parte a acestei versiuni, se potrivea cu implementarea Netscape din Navigator 2.0, ceea ce a însemnat că lui Internet Explorer îi lipseau unele dintre noile obiecte JavaScript, precum acelea reprezentând miniaplicații și imagini.

Aceasta a însemnat continuarea incompatibilităților între cele două versiuni de JavaScript și, prin extensie, de DOM. VBScript a urmat o implementare DOM compatibilă cu versiunea Microsoft de JavaScript.

OBSERVAȚIE

Este dificil de spus încotro se îndreaptă VBScript. Chiar dacă este în dotarea multor utilizatori doar pentru că este un produs Microsoft, mult mai mulți nu apelează la el deoarece Microsoft este singura companie care îl implementează curent în browserul său (cu toate că un program plug-in VBScript este disponibil pentru Netscape). VBScript pare să se îndrepte spre un rol minor până când va fi acceptat de mai multe browsere. Dacă alegeți între limbaje și doriți să căpătați o audiență sporită, folosiți JavaScript.

VECHIUL JAVASCRIPT, NOUL ECMASCRIPT

La începutul lui 1997, Netscape a predat JavaScript Asociației Europene a Producătorilor de Calculatoare (European Computer Manufacturer's Association, ECMA) pentru a fi dezvoltat ca un limbaj standardizat. ECMA a răspuns în vara anului 1997 prin ECMAScript – JavaScript într-o formă standard.

ECMAScript a fost destinat, la origine, să fie un limbaj de scriptare pentru Web. Totuși, standardul ECMAScript prevede posibilități principale de scriptare pentru o varietate de medii gazdă. Din acest motiv, specificația nu leagă limbajul de nici un mediu gazdă particular.

În prezent, Netscape și Microsoft promit amândouă un suport deplin pentru standardul ECMAScript, ceea ce ar putea oferi celor două browsere un teren comun pentru dezvoltarea aplicațiilor de scripturi și DHTML (HTML dinamic). Vor fi ele 100% compatibile? Improbabil. Ca multe alte standarde tehnice, ECMAScript include avertismentul „Este permisă o implementare conformă cu ECMAScript pentru a suporta sintaxa de program nedescrisă în specificație". Și lumea continuă să se învârtă.

Cu Internet Explorer 4.0, Microsoft a oprit încercarea de a se juca de-a prinselea în ceea ce privește modelul de obiect JavaScript, prin extinderea DOM astfel încât să includă fiecare

etichetă și conținut de pe pagină. Noul DOM pe care Microsoft îl oferă este accesibil pentru JavaScript sau VBScript și nu doar pasat către secțiunile cu foaie de stil ale paginii. El include un obiect nou, numit `all`, care este într-adevăr un tablou al tuturor elementelor de pe o pagină (vezi Figura 27.3).

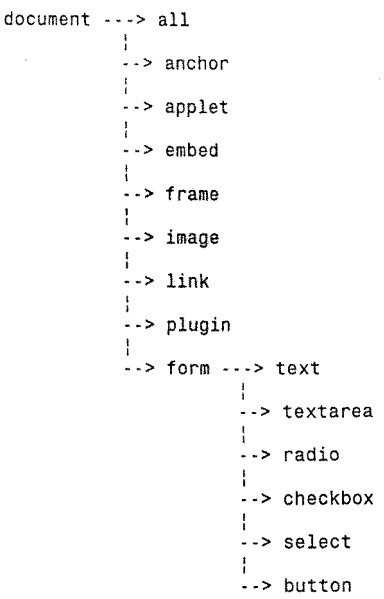


FIGURA 27.3. *Microsoft DOM arată asemănător cu DOM de la Netscape, dar oferă acces la fiecare element prin proprietatea* `all` *a documentului.*

Celălalt lucru pe care DOM de la Microsoft îl face și cel de la Netscape nu îl face este deschiderea lumii *evenimentelor* pentru fiecare element. Evenimentele reprezintă modul în care un browser detectează și răspunde evenimentelor de utilizator, cum ar fi mișcările de mouse sau acțiuni de la tastatură, și sunt tratate mai în detaliu în capitolul următor, „Lucrul cu utilizatorul – Evenimente". În versiunea Netscape de DOM, evenimentele se restrâng la obiecte specifice. De exemplu, singurele evenimente care sunt afectate de executarea clicurilor sunt hiperlegăturile și butoanele. Însă modelul Microsoft deschide toate elementele de pagină pentru, practic, toate evenimentele. De exemplu, puteți detecta execuția unui clic de utilizator pe un titlu sau deplasarea mouse-ului peste un citat bloc.

Folosind modelul IE4, puteți accesa fiecare atribut dintr-un element, incluzând configurări care afectează dimensiunea, culoarea, fontul, fundalul și orice altceva ce este permis pentru eticheta de pagină respectivă. Apoi, dacă mai facem un pas în descrierea modelului, DOM de la Microsoft vă permite, de asemenea, examinarea și modificarea conținutului între seturile de etichete.

Abordarea Microsoft este desigur pe aceeași linie cu lista W3C de cerințe DOM, chiar dacă rămâne o soluție Microsoft, nu un standard. De departe, Microsoft este singura companie care include acest tip de DOM în browserul său. Este bine de lucrat cu el și de văzut cum să îl accesezi folosind scripturi, dar ultimul cuvânt nu depinde de el.

În căutarea unui standard revizuit

Ultima judecată în dezvoltarea unui standard universal este în mâinile Consorțiului World Wide Web. Ceea ce decide W3C va fi adoptat, probabil, în porțiuni importante, dacă nu în întregime, de Netscape, Microsoft și alți producători de browsere.

Secțiunile următoare aruncă o privire către ceea ce W3C a identificat drept cerințe cheie ale unui model DOM universal, punându-se accent pe părțile care vor fi adresate în specificația de nivel 1.

Cerințe generale

Cum ați aflat de mai multe ori, o componentă cheie a modelului DOM de la W3C este că ar trebui să fie universal: utilizatorilor nu ar trebui să li se pretindă să folosească un browser sau un calculator anume pentru a beneficia de avantajul DOM. DOM ar trebui să fie, de asemenea, independent de limbaj. Amintiți-vă că versiunea Netscape și-a început viața legată la JavaScript. Ce dorește W3C este să separe DOM de limbaje, astfel încât un limbaj să poată accesa proprietățile obiectului `document`, dar să nu îi dicteze structura.

O altă caracteristică importantă este legată de angajamentul W3C de creare a accesibilității pentru toți utilizatorii din Web; de aceea, nu se cere o interfață grafică de utilizator pentru conformarea la DOM.

OBSERVAȚIE

Mai multe browsere negrafice sunt disponibile curent, cel mai notabil fiind Lynx (numai text). Alte versiuni sunt destinate persoanelor oarbe sau cu alte infirmități care le împiedică să folosească ceea ce tindem să credem a fi „un browser tipic", cum ar fi interfața grafică de Navigator sau Internet Explorer. Browserele cu caracter special pot interpreta HTML ca vorbire, Braille sau tipuri speciale de caractere.

Capacitatea de a lucra în ambele medii, grafic și negrafic, poate crea unele confuzii producătorilor. Această secțiune explorează utilizarea DOM pentru a crea efecte grafice. Dacă DOM este utilizat pentru atât de mult comportament grafic, de ce să intrăm în bucluc spunând că nu reclamă o interfață grafică de utilizator? Deoarece nu oricine poate folosi o interfață grafică de utilizator, în primul rând oamenii cu dificultăți de vedere. W3C s-a dedicat standardelor care promovează accesibilitatea HTML și DOM pentru cât de multe persoane.

Ultima cerință generală este ca DOM să nu expună browserul utilizatorului la probleme cu securitatea, legalitatea sau proprietatea. Această cerință a generat deja unele discuții în sânul grupului care lucrează la recomandarea DOM. Cu toate că o parte din DOM menționează că toate atributele de pe o pagină trebuie să fie accesibile și modificabile, această cerință poate duce la probleme pentru porțiuni ale modelului de obiect, în special când se referă la agentul de utilizator și la lista de istoric.

Unele pagini sunt destinate să lucreze cu browsere specifice. Browserele se identifică prin atributul `user_agent`, pe care scripturile l-ar putea accesa și modifica. Designerii de pagini lipsiți de scrupule ar putea folosi acest atribut pentru a schimba modul în care un browser vizualizează o pagină astfel încât să pară incompatibil. Rezultatul ar putea duce la o interpretare incorectă a paginii sau ar putea defecta browserul.

Cealaltă problemă apare cu obiectul `history`. De ce ar fi accesul la lista de istoric o chestiune de securitate? Ea depinde de siturile pe care le parcurgeți. Paginile Web cu formulare care cer informații cum ar fi un nume de utilizator, parolă, informații despre cartea de credit sau alte informații particulare, trec adesea acele informații către serverul înglobat într-o adresă URL.

Obiectul `history` ar putea oferi acelora cu intenții mai puțin onorabile încă un mod de a culege informații private pentru un câștig ilicit. Din acest motiv Microsoft și Netscape au dezactivat sau plănuiesc să modifice în vreun fel această parte din DOM.

Navigarea în structură

Acest set de posibilități DOM se referă la moduri de deplasare în interiorul unui document, cum ar fi găsirea părintelui unui element sau a ce copii sunt conținuți într-un element.

PĂRINȚI ȘI COPII ÎN HTML

Ce înseamnă un părinte și un copil? Mergeți să vă întrebați mama și tatăl, această discuție este dincolo de scopul acestei cărți. Dincolo de păsări și albine, relația părinte-copil privește etichetele imbricate. Priviți următorul fragment de cod HTML:

```
<DIV id=parent1>This is Division 1.</DIV>
<DIV id=parent2>This is Division 2 which leads to
<BLOCKQUOTE id=child1>block quote 1 which ends with</BLOCKQUOTE>
the end of Division 2.</DIV>
```

Prima diviziune, `parent1`, nu este un părinte al diviziunii a doua, `parent2`. Ea este de sine stătătoare, deoarece eticheta de închidere a primeia precede eticheta de deschidere a celei de a doua. Citatul bloc, `child1`, este un copil al celei de a doua diviziuni, `parent2`. El stă complet în interiorul limitelor etichetelor de deschidere și închidere a diviziunii.

Iată o altă relație părinte-copil, pe care ar trebui să o recunoașteți:

```
<OL>
  <LI>Item 1</LI>
  <LI>Item 2</LI>
  <LI>Item 3</LI>
</OL>
```

Fiecare dintre articolele listei este un copil al etichetei de listă ordonată, care face articolele listei frate unul cu altul. Dacă ați imbricat o altă listă prin adăugarea unui nou container listă cu propriul său set de articole, noua listă va fi un copil al listei exterioare și un frate pentru celelalte articole ale listei:

```
<OL id=outerlist>
  <LI>Item 1</LI>
  <LI>Item 2</LI>
  <LI>Item 3</LI>
  <UL id=innerlist>
      <LI>Sub item 1</LI>
      <LI>Sub item 2</LI>
      <LI>Sub item 3</LI>
  </UL>
</OL>
```

Folosind notația JavaScript, iată un mod în care ați putea exprima structura fragmentului anterior ca o serie de obiecte:

```
document.outerlist
document.outerlist[0] //Item 1
document.outerlist[1] //Item 2
document.outerlist[2] //Item 3
```

(continuare)

PĂRINȚI ȘI COPII ÎN HTML – CONTINUARE

```
document.outerlist.innerlist
document.outerlist.innerlist[0] //Sub item 1
document.outerlist.innerlist[1] //Sub item 2
document.outerlist.innerlist[2] //Sub item 3
```

Structura notației reflectă structura documentului, reclamând o referință la lista exterioară pentru a ajunge la lista imbricată, interioară. Atunci, fiecare frate există într-o relație paralelă cu vecinul său.

Relațiile părinte-copil din HTML fac mai multe lucruri pentru dumneavoastră. Mai întâi, vă permit să vă referiți la etichete prin relația lor una față de alta, cum se conturează în textul suplimentar „Părinți și copii în HTML". Prin definirea unui mod specific de a identifica elemente în relația lor cu documentul și unul cu altul, vi se oferă o metodă de exprimare a structurii unice a documentului.

După ce fiecare element este identificat și exprimat ca parte a structurii documentului, sunt incluse unele cerințe suplimentare de navigare în structură, inclusiv atribute de expunere și etichete necunoscute.

Manevrați etichetele necunoscute la fel cum ați face cu alte elemente. Dacă are un `id`, ați putea folosi acea valoare să vă referiți la ea în interiorul DOM. Dacă nu, o puteți identifica prin poziția relativă față de alte elemente.

Manipularea documentului și a conținutului

Manipularea documentului și a conținutului urmează standardele de navigare în structură. Manipularea documentului și a conținutului sunt utilizate cu modelul de eveniment pentru a asigura „dinamicul" din „HTML dinamic".

Mai întâi, manipularea documentului vă permite să modificați etichete. De exemplu, ați putea modifica un titlu `H1` într-un titlu `H3` sau într-un paragraf nou cu `P`. În al doilea rând, ea vă permite să adăugați sau să eliminați etichete din document. Partea dificilă este că adăugând sau eliminând etichete de pagină ar trebui să rezulte un document încă valid. De aceea, DOM ar trebui să aibă un mod de a vă opri de la a transforma un set de etichete `LI` în orfani prin eliminarea etichetei părinte `UL` sau `OL` sau de a șterge etichetele `LI` în același timp. W3C nu a dat vreo indicație despre ce metodă preferată va exista în această ediție.

Cu manipularea de conținut, designerul de pagini poate crea scripturi pentru modificări de conținut al oricărei părți a documentului. Această cerință DOM are două părți. Mai întâi, asemănător navigării în structură, ar trebui să fiți capabil să determinați care element înconjoară o bucată particulară de text sau să determinați textul din interiorul oricărui element dat.

După ce sunteți capabil să faceți trimiteri la conținut, puteți adăuga, modifica sau șterge conținutul după dorință. Microsoft a adăugat această trăsătură browserului său Internet Explorer 4.0 cu proprietatea `innerText`. De exemplu, având un titlu `H1` numit `mainHeading`, puteți determina conținutul său în acest mod:

```
var textContent;
textContent = document.mainHeading.innerText;
```

Acest cod creează un șir nou, numit `textContent`, care conține acum tot textul dintre etichetele identificate ca `mainHeading`. Pentru a schimba acest conținut, folosiți aceeași proprietate:

```
textContent = "This is a new heading";
document.mainHeading.innerText = textContent;
```


Suplimentar conținutului, puteți folosi proprietatea pentru a adăuga noi etichete sau elemente. Ați putea schimba linia precedentă în următoarea:

```
textContent = "<H2>This is a new heading</H2>";
document.mainHeading.innerText = textContent;
```

Deoarece afectați numai conținutul, elementele care înconjoară conținutul nu se modifică. Adăugați noul conținut în interiorul etichetelor existente, rezultând codul HTML care arată astfel:

```
<H1 id=mainHeading><H2>This is a new heading</H2></H1>
```

Dacă doriți să înlocuiți etichetele `<H1>` cu `<H2>`, puteți folosi proprietatea `outerText`, care întoarce șirul în interiorul elementului `mainHeading`, inclusiv elementele înseși. Codul ar arăta astfel:

```
TextContent = document.mainHeading.outerText;
textContent = "<H2 id='mainHeading'>This is a new heading</H2>";
document.mainHeading.outerText = textContent;
```

Iar codul HTML rezultat ar arăta astfel:

```
<H2 id=mainHeading>This is a new heading</H2>
```

Deoarece DOM are o varietate de moduri de a examina și modifica documentul dumneavoastră, el necesită și un mod de a afla ce face utilizatorul, astfel încât să poată reacționa adecvat; acest subiect este tratat în următoarea secțiune.

Modelul de eveniment

Pentru a permite documente complet interactive, modelul de eveniment trebuie să includă o largă varietate de activități posibile pentru utilizator, ca deplasări de mouse, clicuri de mouse, combinații de deplasare și clic de mouse (trage și plasează, drag-and-drop), apăsări de taste și acțiuni care gravitează în jurul formularelor. Mai multe informații despre evenimente și subrutine de tratare a evenimentelor există în Capitolul 28.

În versiunile Netscape de modele de obiect, evenimentele specifice erau legate de obiecte specifice. Puteați detecta utilizatorul mișcând indicatorul de mouse peste o hiperlegătură, dar nu peste un titlu; puteați detecta un clic pe un buton, dar nu pe un citat bloc. O parte cheie a cerințelor W3C pentru DOM este ca fiecare element să accepte fiecare eveniment apli- cabil. Orice bucată de text va fi aptă să răspundă activităților mouse-ului, incluzând clicuri, clicuri duble și mutarea indicatorului peste și în afara zonei textului.

Această cerință produce și un oarecare grad de dificultate totuși când este asociată cu probleme de accesibilitate a utilizatorului, de asemenea ridicate de W3C. Ce se întâmplă dacă interfața de utilizator nu acceptă un mouse? Aceasta elimină multe evenimente potențiale din șirul de posibilități pentru o pagină când este vizualizată pe un browser neconvențional. Această problemă nu a fost, încă, abordată în cerințe, dar vă puteți aștepta ca producătorii de browsere să furnizeze alte evenimente, precum apăsarea unei taste anume ca înlocuitor pentru activitățile mouse-ului.

O idee nouă, prevăzută pentru HTML în modelul de eveniment este un concept cunoscut drept *barbotarea evenimentului (event bubbling)*, care recunoaște că un eveniment care afectează un eveniment poate afecta și părinții săi. Prin construcții de scripturi, subrutina de tratare a evenimentului poate lăsa sistemul să cunoască dacă evenimentul a fost tratat complet. Dacă nu a fost, evenimentul este trecut în sus, următorului nivel al documentului.

Spre exemplu, utilizatorul execută clic pe un articol listă ordonată, iar rutina de tratare a evenimentului pentru articolul listă își face treaba. Dacă rutina de tratare a evenimentelor

s-a terminat, iar evenimentul a fost complet îndeplinit, rutina de tratare a evenimentelor returnează valoarea adevărat și evenimentul este complet. Dacă totuși rutina de tratare a evenimentelor returnează valoarea fals, evenimentul este trecut către elementul părinte. Containerul listă ordonată are, acum, posibilitatea să execute acțiuni suplimentare pentru a completa evenimentul. Dacă el returnează valoarea adevărat, evenimentul s-a terminat. Dacă returnează valoarea fals, evenimentul este trimis documentului pentru oricare tratare prestabilită dictată de browserul utilizatorului.

Rezumat

Pe nivelul său cel mai simplu, Modelul de obiect document este un mod de a defini cum pot exprima browserele structura și conținutul lor, indiferent de platformă sau limbaj de scriptare. El își are rădăcinile în Modelul de obiect JavaScript de la Netscape, introdus în Navigator 2.0. Modelul de început de la Netscape a inclus un număr limitat de obiecte și evenimente care prevedeau acces la unele dintre componentele cheie de pe o pagină, precum formulare și hiperlegături, dar ignorau aproape toate celelalte etichete. Ideea originală de la Netscape este acum extinsă ca scop și capacități în Internet Explorer 4.0 de la Microsoft, care include un model de obiect mai apropiat în intenție de cerințele schițate de W3C.

Aceasta nu înseamnă că soluția Microsoft este ultimul cuvânt asupra Modelului de obiect document. Cerințele W3C nu sunt standarde; ele sunt numai o listă de posibilități așteptate de la DOM. Problema conformării la acest soi de listă este că ea nu a creat un standard; ea reprezintă doar o listă de dorințe.

După ce un standard se va dezvolta, va lua, probabil, încă un an producătorilor de software să își modifice browserele pentru a le face să lucreze sigure cu curajoasa lume nouă previzionată de W3C.

Până una alta, fiți cu ochii atât pe Netscape, cât și pe Microsoft. Reviziile lor viitoare de browser sunt un indicator bun al drumului pe care va flutura drapelul W3C, deoarece ambele companii sunt parte a comitetului W3C care lucrează asupra chestiunii. Chiar dacă ele nu îndeplinesc exact modelul DOM final de la W3C, veți fi suficient de aproape pentru a face o tranziție ușoară când ambele browsere se vor apropia de compatibilitatea cu celălalt.

Capitolul 28
Lucrul cu utilizatorul – Evenimente

de Bruce Campbell

BIBL. CENTR. UNIV. „M. EMINESCU" IAȘI
INFORMATICĂ

În acest capitol

- Evenimente intrinseci 399
- De ce se standardizează evenimentele? 400
- Cum lucrează evenimentele 400
- Stabilirea domeniului de existență al subrutinei de tratare a evenimentelor 402
- Lista evenimentelor HTML 4 404
- Alte evenimente de interes 408
- Exemple de utilizare a evenimentelor intrinseci 409

Evenimente intrinseci

O întărire importantă a standardului HTML este adăugarea *evenimentelor intrinseci* la documentele HTML și agenții de utilizator compatibili cu HTML 4. Chiar dacă tratarea evenimentelor în paginile Web înseamnă că trebuie să scrieți cod pentru a crea documente HTML, funcționalitatea evenimentelor de utilizator deschide documentele Web către posibilități infinite de conținut nou și flexibil. Prin adăugarea unui model de eveniment standard, paginile Web devin dinamice și mai sensibile la interacțiunea cu utilizatorul. Planul de lucru curent al Consorțiului World Wide Web (W3C) standardizează 18 evenimente pe care dumneavoastră, ca autor de pagină Web, le puteți utiliza să scrieți scripturi pentru a trata și a crea pagini Web interactive.

OBSERVAȚIE

Amintiți-vă că W3C folosește termenul *agent de utilizator (user agent)* ca un superset al browserelor Web. Cuvântul *browser* evocă cerința de a vedea pentru a utiliza aplicația. W3C dorește ca agenții de utilizator să fie creați pentru fiecare, inclusiv pentru cei care au deficiențe de vedere. Folosesc termenul agent de utilizator adesea în acest capitol.

Documentele HTML 4 bazate pe Web descarcă o pagină sofisticată Web la un client o singură dată, într-o stare inițială. După ce documentul termină încărcarea în agentul de utilizator, o pagină Web HTML 4 se poate modifica dinamic fără reîntoarcere în Web pentru a cere informații adiționale, ceea ce reduce cererea de pagini a serverului Web în mod semnificativ. Evenimentele HTML 4 dau impulsul pentru schimbări dinamice. Când are loc un eveniment, agentul de utilizator recunoaște evenimentul și procesează lista de evenimente intrinseci HTML 4 în detaliu. Cuvântul *intrinsec* se referă la agentul de utilizator care este deja conștient de evenimente, fie că un autor de pagină Web le folosește în mod dinamic, pentru a schimba pagini Web, fie că nu.

Da, pentru a folosi evenimentele intrinseci ale specificației HTML 4, va trebui să scrieți cod de scriptare și să includeți codul în documentele dumneavoastră HTML sau în fișiere

adiționale la care faceți trimitere din documentele dumneavoastră HTML. Detaliile adăugării de scripturi paginilor dumneavoastră Web sunt tratate în Capitolul 29, „HTML și scripturi – Combinația DHTML". Acest capitol, unul de pregătire pentru capitolul 29, acoperă promisiunea de pagini Web dinamice făcute posibile de o listă de evenimente standard. Într-un browser Web static, adică unul dintre acelea conforme strict cu specificația HTML 3.2, nu există nici o modalitate pentru un autor Web să prevadă conținut dinamic fără întoarcere la serverul Web și încărcarea unei alte pagini Web. Serverele Web trebuie să trateze mai multe cereri de pagini în modelul static de document. Lista de evenimente intrinseci pentru HTML 4 este rampa de lansare pentru a adăuga funcționalitate dinamică și interactivă paginilor dumneavoastră Web.

De ce se standardizează evenimentele?

Dacă fiecare producător al unui agent de utilizator inventează o listă proprie de evenimente, un autor de pagini Web va trebui să scrie cod specific pentru fiecare agent de utilizator. S-ar putea ca doi producători să prevadă aceeași funcționalitate, dar, dacă ei dau nume diferite evenimentelor lor, conținutul pe care l-ați creat pentru un agent de utilizator ar putea să nu funcționeze pe celălalt.

Motivele standardizării listei de evenimente sunt, în principal, aceleași ca și pentru standardizarea numelor de etichetă și de atribut: standardizare în rutina de analiză a documentului HTML din interiorul agentului de utilizator și consecvență în setul de caracteristici minimale sigure al agenților de utilizator compatibili.

Cum lucrează evenimentele

Pentru ca evenimentele să fie permise pe o pagină Web, se cere o relație puternică de simbioză între producătorul agentului de utilizator și autorul paginii Web. Specificația HTML 4 standardizează relația agent de utilizator – pagină Web. Producătorul scrie cod în interiorul agentului de utilizator, care supraveghează dacă are loc oricare eveniment dintr-o listă finită de evenimente, în timp ce vizitatorul de sit folosește aplicația agent de utilizator. Programatorii se referă la obiectul cod care recunoaște evenimentul ca la o *rutină de ascultare a evenimentelor (event listener, ascultător de evenimente)*.

Pe de altă parte, autorul paginii Web revede lista evenimentelor disponibile și scrie cod care face să se întâmple ceva pentru fiecare eveniment pe care autorul dorește să îl activeze pe pagina Web. Codul pe care autorul Web îl scrie se numește *rutină de tratare a evenimentelor (event handler, manevrant de evenimente)*. Ca autor, puteți lega fiecare rutină de tratare a evenimentelor de oricât doriți din pagina Web. De exemplu, pe calculatoarele din zilele noastre, cele mai multe aplicații tratează un eveniment de executare a unui clic de mouse ori de câte ori un utilizator execută clic pe o aplicație activă. HTML 4 nu face excepție. În această privință, puteți scrie rutine de tratare a evenimentelor separate pentru a trata clicurile pe imagini diferite din pagina dumneavoastră Web sau pe oricare element HTML vizibil sau puteți considera un singur eveniment pentru întregul document HTML. De asemenea, rutinele de tratare a evenimentelor au un domeniu de valabilitate, pe care veți ști să-l definiți când veți termina de citit acest capitol.

Partea creativă a procesului este să decideți ce să faceți în interiorul rutinelor de tratare a evenimentului. Când un utilizator execută clic pe o imagine din pagina dumneavoastră Web, puteți modifica imaginea, extinde un paragraf sau un text, modifica atributele de text într-o altă parte a paginii, porni o animație, merge oriunde în Web pentru a obține mai multe informații sau face ceva ce nimeni altcineva nu s-a gândit să facă înainte. Posibilitățile par fără sfârșit, dacă luați în considerare 18 evenimente și zeci de acțiuni posibile pentru fiecare eveniment.

În Capitolul 29, veți vedea exemple mai detaliate de subrutine de tratare a evenimentelor. Acest capitol se concentrează mai mult pe configurarea evenimentelor în mod adecvat în interiorul structurii de etichete a documentelor voastre HTML 4. Veți inițializa evenimentele ca atribute de etichete HTML prin folosirea numelui standard de eveniment ca nume de atribut și furnizarea unei valori, care este numele funcției de scriptare potrivită. De exemplu, pentru a activa un eveniment clic pentru o imagine pe pagina dumneavoastră Web, iată cum specificați eticheta de imagine:

```
<IMG NAME="image1" SRC="image1.gif" onClick="Function1();">
```

Apoi, scrieți o rutină de scriptare numită `Function1()`, sau puteți trata evenimentul în chiar interiorul etichetei `<IMG>` prin includerea codului potrivit ca valoare pentru atributul `onClick`. Tratarea evenimentelor complet în interiorul unei etichete HTML este numită *tratare de eveniment inline (inline event handling)*. Acest capitol include un exemplu de rutină de tratare a evenimentului inline în secțiunea „Modificări de atribut al textului inline".

Restul acestui capitol privește modul în care autorii pun evenimentele la treabă. Până să vă cufundați complet în responsabilitățile de creare a paginii Web, luați în considerare toată munca pe care producătorul agentului de utilizator a trebuit să o depună pentru a face evenimentele să lucreze pentru dumneavoastră. Desigur, nu există nici un motiv pentru care nu puteți fi și dumneavoastră producător de agent de utilizator. Iertați-mă pentru presupunerea că plănuiți să creați conținut pentru browserele Web Communicator de la Netscape sau Internet Explorer de la Microsoft, dacă, la fel de bine, vă scrieți propriul dumneavoastră agent de utilizator. După cum ați realizat sigur până acum, scrierea unui browser Web complet compatibil cu HTML 4 este semnificativ mai dificilă decât scrierea unui browser Web complet compatibil cu HTML 1, 2 sau 3.2.

Cele mai multe dintre evenimentele la care un agent de utilizator HTML 4 ascultă ar fi foarte dificil sau imposibil de scris dacă sistemul de operare nu ar prevedea, deja, datele de bază pentru eveniment. Luând ca exemplu sistemele de operare Windows, NT, UNIX și Mac, toate pun clicurile dispozitivelor de indicare la dispoziția aplicațiilor scrise pe respectiva platformă. Sistemele de operare transmit agentului de utilizator cel puțin poziția x și y pentru un clic și care buton a fost apăsat.

Producătorul agentului de utilizator este responsabil pentru maparea datelor brute ale evenimentului de la sistemul de operare la semnificația sa pe pagina Web încărcată. Pentru evenimentul clic, agentul de utilizator trebuie să știe unde a fost plasat fiecare element HTML pe pagina Web și, apoi, să considere cum un clic de mouse din sistemul de operare se translatează la un element de pagină Web. Apoi, agentul de utilizator trebuie să se refere la pagina Web încărcată pentru a vedea implicațiile acelui eveniment. Dacă autorul de pagină Web nu a scris o subrutină de tratare a evenimentelor specifică pentru respectivul eveniment aplicat respectivului element, agentul de utilizator mărește domeniul de valabilitate al evenimentului. Dacă nu există nici o rutină de tratare a evenimentului în interiorul domeniului de existență a respectivului eveniment pe pagina Web, agentul de utilizator poate să abandoneze detaliile evenimentului și să se deplaseze la următorul eveniment din coadă.

OBSERVAȚIE

Prin scrierea unui agent de utilizator în Java, un producător poate beneficia de avantajul mașinii virtuale Java, care face ea însăși munca sistemului de operare specific.

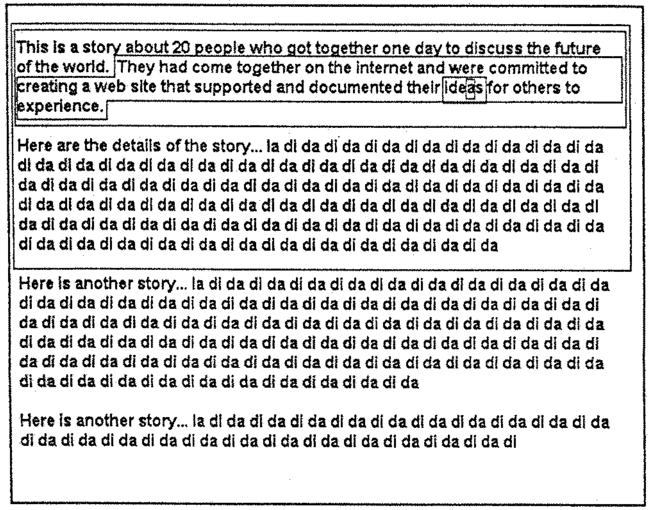
Rutina de ascultare a evenimentelor din agentul de utilizator adaugă continuu noi evenimente în coada de evenimente, iar agentul de utilizator consideră fiecare eveniment din coadă în ordinea sosirii. Pentru fiecare eveniment, agentul de utilizator asigură funcționalitatea potrivită care este specificată de autorul paginii Web. Dacă, la un moment, utilizatorul încarcă altă pagină Web, coada de evenimente este golită și începe acceptarea de noi evenimente pentru pagina nou încărcată. Acum, după ce v-ați familiarizat cu evenimentele standard ale HTML 4, să vedem care sunt implicațiile de executare a muncii evenimentului din perspectiva producătorului agentului de utilizare. Unele evenimente sunt mult mai ușor activate decât altele. Pentru fiecare eveniment adăugat specificației HTML 4, producătorul agentului de utilizator are o seamă de lucruri de făcut.

Stabilirea domeniului de existență al subrutinei de tratare a evenimentelor

Preluând avantajul ierarhiei naturale a paginilor Web, rutinele de tratare a evenimentelor pot fi mai precise în definirea domeniului lor de existență. Paginile HTML 4 au o ierarhie strictă. Un caracter dintr-o pagină Web este parte a unui cuvânt dintr-o pagină Web, care este o parte a unei propoziții, care este parte a unui paragraf, care poate fi parte a unei diviziuni, care este o parte a întregului document.

Figura 28.1 demonstrează ierarhia elementelor pe o pagină Web. Folosind perechea de etichete `<SPAN></SPAN>` puteți crea litera `a` ca un element caracter valid, cuvântul `ideas` ca un element cuvânt valid, iar propoziția cu cuvântul *ideas* ca un element propoziție valid. Paragraful cu cuvântul *ideas* este un element paragraf valid pe care îl puteți crea folosind perechea de etichete `<P></P>`. În final, primele două paragrafe creează un element diviziune valid, dacă este conținut într-o pereche de etichete `<DIV></DIV>`.

FIGURA 28.1.
Ierarhia de elemente HTML.



În termeni de ierarhie, elementul literă *a* este elementul de pe cel mai jos nivel, iar elementul diviziune este elementul de pe nivelul cel mai înalt. Când un utilizator execută clic pe litera *a* în cuvântul *ideas*, el execută, în realitate, clic pe cinci elemente, neincluzând și documentul pagină Web însuși, dacă toate etichetele au fost incluse cum se sugerează în paragraful precedent.

Puteți activa rutinele dumneavoastră de tratare a evenimentelor din interiorul oricărei perechi de etichete pentru care acel eveniment este valid. Stabiliți domeniul de existență potrivit bazându-vă pe eticheta în care plasați evenimentul ca un atribut. De exemplu, considerați următorul fragment de document HTML:

```
<DIV>
  <IMG SRC="image1.gif">
  <IMG SRC="image2.gif">
  <IMG SRC="image3.gif">
</DIV>
```

Acest cod creează o diviziune cu trei imagini plasate în interiorul perechii de etichete de diviziune. Puteți activa evenimentul `onClick` pentru o anume imagine prin plasarea unui atribut `onClick` în eticheta `<IMG>`, cum se arată în continuare:

```
<IMG SRC="image1.gif" onClick="RunMe();">
```

Sau puteți plasa atributul `onClick` în interiorul etichetei `<DIV>` pentru a activa un singur eveniment `onClick` pentru toate trei imaginile:

```
<DIV onClick="RunMe();">
```

Dacă plasați un atribut `onClick` în interiorul etichetei `<IMG>` și eticheta de deschidere `<DIV>`, atunci agentul de utilizator va folosi numai atributul celui mai jos element din ierarhia de elemente a paginii Web. În acest caz, atributul `onClick` al etichetei `<IMG>` va fi activat, dacă există, înainte de luarea în considerație a atributului `onClick` al etichetei de deschidere `<DIV>`. O dată ce un agent de utilizator găsește un eveniment activat într-o etichetă de nivel jos, el nu consideră același atribut de eveniment în etichetele de nivel mai înalt.

OBSERVAȚIE

Microsoft numește procesul prin care agentul de utilizator caută, mai întâi în etichetele cele mai joase și apoi se mută în susul ierarhiei de elemente *barbotarea evenimentului (event bubbling)*. În Internet Explorer 4.0, puteți menționa ca agentul de utilizator să nu se uite spre etichetele mai înalte prin contramandarea barbotării de evenimente. Totuși, la data acestei scrieri, mecanismul de contramandare a barbotării evenimentelor nu este standardizat în ultima schiță de lucru HTML 4 de la W3C.

Pentru multe evenimente standard HTML 4, puteți specifica o subrutină de tratare a evenimentelor pentru document ca un întreg. Agentul de utilizator HTML 4 asociază pagina Web activă cu un obiect `document`. Puteți gândi obiectul `document` ca pe obiectul de nivelul cel mai înalt al fiecărei pagini Web în ierarhia de evenimente. Când scrieți o rutină de tratare a evenimentelor pentru document, dumneavoastră garantați că o rutină de tratare a evenimentului va fi rulată pentru acel eveniment specific ca un ultim resort de către agentul de utilizator. Obiectul `document` vă oferă și o oportunitate de a simplifica paginile dumneavoastră Web prin scrierea unei singure rutine, în locul unor rutine separate pentru fiecare element de pe pagină. Dacă aveți nevoie de rutine specifice pentru anumite documente, plasați atributele potrivite între respectivele etichete. Și în acest caz puteți crea rutina de tratare a evenimentului `document` astfel încât să trateze evenimentele în legătură cu toate celelalte elemente. Capitolul 29 explică lucrurile specifice pentru crearea rutinelor de tratare a evenimentului `document`.

Lista evenimentelor HTML 4

Acum, că stăpâniți marele spectacol al evenimentelor, rutinelor de tratare a evenimentelor și rutinelor de ascultare a evenimentelor, este timpul să priviți către aspectele specifice ale evenimentelor standardizate de către ultima schiță de lucru W3C asupra HTML 4. Există 18 evenimente de considerat, listate aici:

onload	onmouseover	onkeydown
onunload	onmousemove	onkeyup
onclick	onmouseout	onsubmit
ondblclick	onfocus	onreset
onmousedown	onblur	onselect
onmouseup	onkeypress	onchange

Următoarea secțiune explică fiecare eveniment din listă.

Evenimentul onload

Evenimentul onload are loc atunci când agentul de utilizator termină încărcarea unei ferestre sau a tuturor cadrelor într-un FRAMESET. Acest atribut poate fi utilizat cu elementele BODY și FRAMESET.

Evenimentul onload vă oferă o ocazie minunată de a crea pagini Web care variază bazându-se pe informații care se modifică în altă parte în Web sau pe informații curente de pe serverul dumneavoastră Web. Cu evenimentul onload, puteți scrie o subrutină de tratare a evenimentelor care așază la un loc pagina Web piesă cu piesă din surse dinamice, precum un server de bursă, server de timp probabil sau server de știri. Evenimentul onload se declanșează numai o dată per pagină Web, deci agentul de utilizator nu trebuie să ajusteze continuu efectele rutinei sale de scriptare. Ar mai trebui să rețineți că evenimentul onload pornește fără nici o altă activitate specifică din partea utilizatorului.

Evenimentul onunload

Evenimentul onunload are loc atunci când agentul de utilizator șterge un document dintr-o fereastră sau dintr-un cadru. Acest atribut poate fi utilizat cu elementele BODY și FRAMESET.

Evenimentul onunload este parte a listei de evenimente HTML 4 pentru a permite o procesare sofisticată între un server Web și aplicația activă a agentului de utilizator. Evident, utilizatorul nu observă imediat efectul rutinei de tratare a evenimentului onunload, deoarece pagina nu mai este activă în agentul de utilizator. Totuși, pagina ar putea fi alterată în memoria cache în caz că este vizitată din nou, ori serverul ar putea fi conștientizat că utilizatorul nu mai este pe pagina respectivă.

Într-o aplicație de pagină Web integrată care interconectează mai mulți utilizatori, evenimentul onunload ar putea fi făcut să informeze alți utilizatori că, în momentul de față, un utilizator nu mai participă în aplicația de pagină Web respectivă. Pentru un utilizator singular, evenimentul onunload ar putea hotărî modificări în alt cadru al unui set de cadre.

Evenimentul onclick

Evenimentul onclick are loc atunci când butonul dispozitivului de indicare este apăsat deasupra unui element. Acest atribut poate fi utilizat cu majoritatea elementelor.

Evenimentul onclick va fi, probabil, cel mai des folosit. El transformă un utilizator Web într-un participant activ, în locul unui cititor pasiv de informație. Aproape toți utilizatorii de calculator au un dispozitiv de indicare la dispoziție. Cu evenimentul onclick, puteți face să aibă loc modificări dinamice pe o pagină Web ori de câte ori un utilizator execută clic undeva pe ea.

Evenimentul ondblclick

Evenimentul ondblclick are loc atunci când, cu butonul dispozitivului de indicare, se execută dublu clic deasupra unui element. Acest atribut poate fi utilizat cu majoritatea elementelor.

Evenimentul ondblclick deschide un element de pagină Web către posibilități de eveniment multiple. Un utilizator poate distinge între clicurile singulare și duble pentru a interacționa cu o pagină Web. Autorul de pagină Web poate scrie două rutine de tratare a evenimentului separate bazându-se pe același element. Un clic dublu este mai dificil de efectuat de utilizator, așa că ar fi bine să se atribuie rutinei de tratare a evenimentului ondblclick o opțiune selectată mai rar, iar evenimentului onclick una mai populară.

Uzual, luați în considerare evenimentul ondblclick când deja există o rutină de tratare a evenimentului onclick pentru un element sau când intenționați ca evenimentul să fie declanșat numai de utilizatorii experimentați, de la care așteptați să fie instruiți în utilizarea paginii dumneavoastră Web. Surferii din Web sunt în stare să meargă executând clic pe fiecare parte a unei pagini Web, dar e mai puțin probabil să meargă executând dublu clic doar pentru amuzament.

Evenimentul onmousedown

Evenimentul onmousedown are loc atunci când butonul dispozitivului de indicare este apăsat peste un element. Acest atribut poate fi utilizat cu majoritatea elementelor.

Ca și evenimentul onclick, evenimentul onmousedown implică o apăsare a utilizatorului pe un buton de dispozitiv de indicare. Totuși, evenimentul onmousedown permite modificări ale paginii Web dinamice, care sunt active numai când butonul dispozitivului de indicare este apăsat; aceste modificări pot fi inversate prin scrierea unei rutine de tratare a evenimentului separată pentru evenimentul onmouseup.

Evenimentul onmousedown poate fi folosit cu activități de tragere și plasare (drag-and-drop) pentru a declanșa un eveniment când utilizatorii încep operația de tragere.

Evenimentul onmouseup

Evenimentul onmouseup are loc atunci când butonul dispozitivului de indicare este eliberat deasupra unui element. Acest atribut poate fi utilizat cu majoritatea elementelor.

Ca și evenimentul onclick, evenimentul onmouseup implică eliberarea de către utilizator a unui buton de dispozitiv de indicare, dar permite și modificări ale paginii Web dinamice, care nu sunt active până când butonul dispozitivului de indicare nu este eliberat.

Evenimentul onmouseup poate fi folosit cu activități de tragere și plasare (drag-and-drop) pentru a declanșa un eveniment când utilizatorii termină operația de tragere a unui element de pagină deplasabil.

Evenimentul onmouseover

Evenimentul onmouseover are loc atunci când dispozitivul de indicare intră în zona unui element de pagină Web. Acest atribut poate fi utilizat cu majoritatea elementelor.

Folosiți evenimentul onmouseover când doriți să declanșați un eveniment pentru un element de pagină Web fără să fie necesar ca utilizatorul să apese un buton de dispozitiv de indicare. Evenimentul se declanșează de fiecare dată când utilizatorul mută cursorul dispozitivului de indicare peste elementul asociat cu rutina de tratare a evenimentului.

Rutinele de tratare a evenimentului pentru onmouseover prevăd, uzual, mai multe informații despre elementul de interes prin extinderea unei secțiuni sau adăugând informații noi paginii Web. Sau rutina de tratare a evenimentului poate schimba atributul unui element astfel încât să îl accentueze față de alte elemente similare ale unei pagini Web.

Evenimentul onmousemove

Evenimentul onmousemove are loc atunci când dispozitivul de indicare se mișcă peste un element. Acest atribut poate fi utilizat cu majoritatea elementelor.

Evenimentul onmousemove diferă de evenimentul onmouseover prin aceea că el se declanșează de multiple ori când mouse-ul se deplasează peste un element. Evenimentul onmouseover pornește numai o dată la intrarea în suprafața unui element de pagină Web, dar evenimentul onmousemove se declanșează continuu atâta vreme cât dispozitivul de indicare al utilizatorului se mișcă peste un element. Frecvența de declanșare este, într-un fel, condiționată de logica agentului de utilizator în procesarea evenimentului, așa încât fiecare producător poate lua o decizie diferită asupra a cât de des să pornească evenimentul.

Evenimentul onmousemove poate fi folosit cu o activitate de tragere și plasare pentru a actualiza obiectul tras.

OBSERVAȚIE

Asigurați-vă că înțelegeți cum interacționează familia de evenimente onmouse cu familia de evenimente onclick pentru agenții de utilizator pe care îi vizați. Schița de lucru HTML 4 nu spune nimic despre care eveniment pornește primul când o activitate a utilizatorului reprezintă, potențial, unul din două sau mai multe evenimente. De exemplu, executarea unui clic determină atât un eveniment mouse down (mouse apăsat), cât și unul mouse up (mouse eliberat). Dacă includeți două sau mai multe evenimente în interiorul etichetelor unui element, aveți nevoie să cunoașteți aspectele specifice ale agentului utilizator despre care evenimente se declanșează, în ce ordine. Vizitați situl Microsoft (www.microsoft.com) pentru detalii asupra implementării lor în Internet Explorer 4.0.

Evenimentul onmouseout

Evenimentul onmouseout are loc atunci când dispozitivul de indicare este îndepărtat de un element. Acest atribut poate fi utilizat cu majoritatea elementelor.

Folosiți evenimentul onmouseout când doriți să declanșați un eveniment pentru un element de pagină Web fără să fie necesar ca utilizatorul să apese un buton de dispozitiv de indicare. Evenimentul se declanșează de fiecare dată când utilizatorul îndepărtează cursorul dispozitivului de indicare de elementul asociat cu rutina de tratare a evenimentului.

Rutinele de tratare a evenimentului pentru onmouseout prevăd, uzual, să restrângă informații suplimentare despre elementul de interes prin eliminarea unei secțiuni a paginii Web. Rutina de tratare a evenimentului poate schimba atributul unui element astfel încât să îl accentueze față de alte elemente similare de pe o pagină Web, arătând explicit că un element a fost deja vizitat.

Evenimentul onfocus

Evenimentul onfocus are loc atunci când un element este focalizat, fie prin dispozitivul de indicare, fie prin ordinea de tabulare a navigației. Acest atribut poate fi utilizat cu elementele următoare: LABEL, INPUT, SELECT, TEXTAREA și BUTTON.

Teoretic, evenimentul onfocus lucrează similar cu evenimentul onmouseover, pe care îl folosiți adesea cu elemente de text și imagine, dar este folosit în mod specific cu elemente legate de formulare.

Evenimentul onblur

Evenimentul onblur are loc atunci când un element pierde focalizarea, fie prin dispozitivul de indicare, fie prin ordinea de tabulare a navigației. Acest atribut poate fi utilizat cu elementele următoare: LABEL, INPUT, SELECT, TEXTAREA și BUTTON.

Teoretic, evenimentul onblur lucrează similar cu evenimentul onmouseout, pe care îl folosiți adesea cu elemente de text și imagine, dar este folosit în mod specific cu elemente legate de formulare.

Evenimentul onkeypress

Evenimentul onkeypress are loc atunci când o tastă este apăsată și eliberată peste un element. Acest atribut poate fi utilizat cu majoritatea elementelor.

Evenimentul onkeypress este similar evenimentului onclick, doar că el oferă mult mai multe posibilități condiționale. Un dispozitiv indicator poate avea, tipic, trei butoane de apăsat, dar o tastatură poate avea peste o sută de taste distincte. Evenimentul onkeypress se declanșează ori de câte ori un utilizator apasă orice tastă pe o tastatură, chiar și pentru elementul peste care cursorul dispozitivului de indicare este poziționat curent.

Evenimentul onkeydown

Evenimentul onkeydown are loc atunci când o tastă este apăsată peste un element. Acest atribut poate fi utilizat cu majoritatea elementelor.

Evenimentul onkeydown este similar evenimentului onmousedown, doar că el oferă mult mai multe posibilități condiționale deoarece o tastatură are mai multe taste față de câte butoane are un dispozitiv indicator.

Evenimentul onkeydown ia în considerare modificările de pagină Web dinamică care sunt active numai în timp ce tasta este apăsată: aceste modificări pot fi inversate scriind o rutină de tratare a evenimentelor separată pentru evenimentul onkeyup.

Evenimentul onkeyup

Evenimentul onkeyup are loc atunci când o tastă este eliberată peste un element. Acest atribut poate fi utilizat cu majoritatea elementelor.

Evenimentul onkeyup este similar evenimentului onmouseup, doar că el oferă mult mai multe posibilități condiționale deoarece tastatura are mai multe taste față de câte butoane are un dispozitiv indicator.

Evenimentul onkeyup ia în considerare modificările de pagină Web dinamică care nu sunt active decât atunci când utilizatorul eliberează butonul dispozitivului de indicare. Rutina de tratare a evenimentului onkeyup poate inversa modificările dinamice făcute de evenimentul onkeydown.

Evenimentul onsubmit

Evenimentul onsubmit are loc atunci când un formular este transmis. El se aplică numai elementului FORM.

Folosiți rutina de tratare a evenimentului onsubmit să efectuați taskuri bazate pe prezentarea unui formular HTML, cum ar fi schimbarea culorii de fundal a formularului spre a arăta că a fost transmis.

Evenimentul onreset

Evenimentul onreset are loc atunci când un formular este reinițializat. Se aplică numai elementului FORM.

Folosiți o rutină de tratare a evenimentului onreset să efectuați taskuri bazate pe reinițializarea unui formular HTML, cum ar fi schimbarea culorii de fundal a formularului spre a indica faptul că a fost reinițializat.

Evenimentul onselect

Evenimentul onselect are loc atunci când un utilizator selectează text într-un câmp de text. Acest atribut poate fi folosit cu elementele INPUT și TEXTAREA.

Veți scrie rutine de tratare a evenimentului onselect pentru a trata procesarea de formular avansată, câmp cu câmp, și, potențial, pentru a comunica cu o bază de date. Rutina dumneavoastră ar putea să actualizeze informațiile din pagina Web bazându-se pe câmpurile de formular de interes sau să ofere utilizatorului mai multe informații despre datele valide de introdus în respectivul câmp.

Evenimentul onchange

Evenimentul onchange are loc atunci când un control pierde focalizarea (activarea unei aplicații) și valoarea sa a fost modificată până la recăpătarea focalizării. Acest atribut poate fi utilizat cu elementele următoare: INPUT, SELECT și TEXTAREA.

Rutinele de tratare a evenimentului onchange sunt folosite pentru a trata procesarea de formular avansată, câmp cu câmp, și, potențial, pentru a comunica cu o bază de date. Rutina dumneavoastră ar putea să actualizeze informațiile din pagina Web bazându-se pe câmpurile de formular de interes sau să execute dinamic o verificare a validității datelor modificate din respectivul câmp.

OBSERVAȚIE

Veți învăța mai multe despre importanța evenimentelor onselect și onchange în Capitolul 30, „Asocierea datelor".

Alte evenimente de interes

Așa cum au creat și promovat propriile lor etichete HTML și atribute dincolo de schițele de lucru și recomandările W3C, cu siguranță că producătorii de agenți de utilizator vor extinde lista de evenimente dincolo de cea acceptată de W3C.

În prezent, Microsoft a devenit cea mai agresivă în recomandarea de noi evenimente și încorporarea lor în browserul Web Internet Explorer 4.0. Microsoft a aliniat noi evenimente HTML cu modelul de evenimente disponibil în Visual Basic pentru aplicații scrise în Visual Basic. Modelul de evenimente din Visual Basic a reprezentat un mare economizor de timp pentru programatori, deci de ce să nu includem aceleași caracteristici pentru

paginile Web? Într-adevăr, evenimentele HTML 4 fac paginile Web să funcționeze mai mult ca aplicații decât ca pagini gen enciclopedie de informații statice, iar Visual Basic reprezintă o platformă de succes pentru multe aplicații.

De fapt, cele mai multe dintre exemplele din Capitolul 29 funcționează curent numai în Internet Explorer 4.0. Microsoft folosește cele 18 evenimente standard sugerate de W3C în schița sa de lucru HTML 4, dar adaugă, de asemenea, noi evenimente tehnologiei pe care o numește *HTML dinamic*. Lista următoare include 15 evenimente adiționale care pot fi tratate când scrieți cod pentru browserul Web Internet Explorer:

onhelp	onafterupdate	ondatasetcomplete
onscroll	onbeforeupdate	onerrorupdate
onrowenter	onbeforeunload	onfinish
onrowexit	ondatasetavailable	onreadystatechange
onabort	ondatasetchanged	onstart

Evenimentul onhelp se declanșează când un utilizator selectează caracteristica Help în timp ce dispozitivul de indicare este poziționat peste un element. Evenimentul onscroll se declanșează când un utilizator defilează conținutul unui element de pagină Web capabil de defilare sau întregul document pagină Web. Evenimentele onrowenter și onrowexit funcționează asemănător evenimentelor onmouseover și onmouseout, cu excepția faptului că ele sunt specifice elementelor linie de tabel și pot fi declanșate folosind tasta Tab.

Celelalte 11 evenimente din lista anterioară sunt specifice procesării avansate în formulare cu date asociate. De fapt, evenimentele onrwenter și onrowexit funcționează bine, în special cu câmpuri de date într-un tabel creat dinamic. Capitolul 30 acoperă asocierea datelor și descrie aceste evenimente.

Pe măsură ce modelul de evenimente HTML 4 se stabilește ca standard, așteptați-vă ca și Netscape să dezvolte noi evenimente. Vizitați situl Web Netscape pentru detalii la http://www.netscape.com.

Exemple de utilizare a evenimentelor intrinseci

Capitolul 30 prezintă exemple detaliate de rutine de tratare a evenimentelor scriptate care permit evenimentelor intrinseci HTML 4 să creeze pagini Web dinamice. Nu există nici o limită inerentă asupra dimensiunii unei rutine script de tratare a evenimentelor. Rutinele de tratare a evenimentelor pot conține o logică de condiții detaliată și pot apela subproceduri multiple, toate în același fișier HTML în care se găsesc etichetele HTML și textul de rând.

Acest capitol vă oferă un exemplu de rutine de tratare a evenimentelor simple care nu reclamă funcții de scriptare sofisticate. Tratarea evenimentelor în interiorul unei etichete HTML este numită *tratarea evenimentului inline (inline event handling)*. După exemplul de tratare inline, alte exemple vă arată cum să configurați evenimentele pentru unele modificări dinamice interesante de pagină Web. Rutinele script de tratare a evenimentelor nu sunt acoperite de acest capitol, dar veți poseda o mai bună înțelegere a posibilităților înainte să citiți detaliile din Capitolul 29.

Modificări de atribut al textului inline

Pentru a crea efecte de text dinamic, puteți configura evenimentele și modificările de atribut al textului chiar din interiorul etichetelor HTML. Orice modificare dinamică a unei pagini Web pe care o creați exclusiv într-o singură etichetă HTML este numită *modificare inline (inline change)*. Modificările inline sunt foarte simple.

Ca exemplu, aruncați o privire Listingului 28.1, care modifică dinamic o dimensiune de font de titlu ca răspuns la acțiunea utilizatorului. Fișierul HTML este foarte scurt.

Listingul 28.1. O modificare de atribut al textului inline.

```
<HTML>
<HEAD>
<TITLE>Test</TITLE>
</HEAD>
<BODY BGCOLOR=#FFFFFF>
<H1 onmouseover="this.style.fontSize=128"
    onmouseout=" this.style.fontSize=32">
Hello World
</H1>
</BODY>
</HTML>
```

În Listingul 28.1, observați două atribute de eveniment în eticheta de deschidere `<H1>`. Atributul `onmouseover` configurează un eveniment să se declanșeze când cititorii plasează dispozitivele lor de indicare peste cuprinsul titlului. Când dispozitivul de indicare poposește peste titlu, textul titlului se modifică la dimensiunea de 128 de puncte tipografice. Atributul `onmouseout` configurează un eveniment să se declanșeze când cititorii mută dispozitivul de indicare în afara titlului. Când mouse-ul părăsește aria titlului, textul titlului se modifică într-unul mult mai mic, de 32 de puncte.

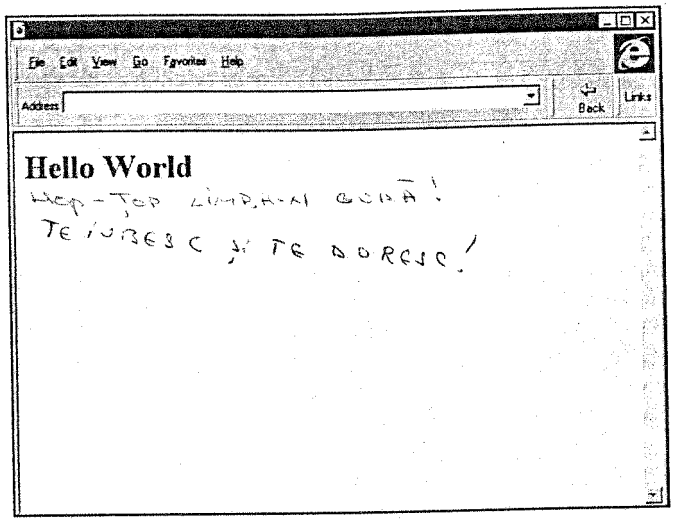
Pentru a schimba în mod dinamic atributele textului într-o etichetă HTML, beneficiați de avantajele caracteristicilor înglobate de modelul de obiect HTML 4. Cuvântul cheie `this` se referă la elementul HTML identificat de eticheta în care este conținut cuvântul cheie. Toate elementele de text au un obiect stil pe care îl puteți modifica în mod dinamic. Deci adăugați numele proprietății de stil potrivite și atribuiți o nouă valoare proprietății. Ați putea modifica culoarea dinamic prin atribuirea unei valori ca `Red` proprietății `this.style.color`.

Figura 28.2 arată pagina Web din Listingul 28.1 după ce utilizatorul a plasat mouse-ul peste titlu.

FIGURA 28.2. *Textul este mărit ca răspuns la evenimentul* `onmouseover`.

Figura 28.3 arată aceeași pagină Web după ce utilizatorul a deplasat mouse-ul în afara titlului Hello World.

FIGURA 28.3. *Ca răspuns la evenimentul* `onmouseout`, *textul titlului se micșorează.*



Tratarea inline este concisă, dar nu deosebit de robustă. Pentru modificări dinamice inline, nu este necesar să creați nici un script pentru paginile dumneavoastră Web, însă scripturile sunt o trăsătură care vă permite să adăugați logică condițională paginilor. Pentru a reduce timpul de descărcare pentru rutine de tratare a evenimentului sofisticate, veți dori, desigur, să includeți codul o singură dată în document. Creați rutina de tratare a evenimentului ca pe un script și referiți-vă la script de mai multe ori din etichete HTML multiple. Capitolul 29 acoperă scriptarea.

Exemple de atribute de eveniment

Următoarele exemple de atribute de evenimente sunt etichete HTML 4 valide care configurează rutinele de tratare a evenimentelor fără prezentarea efectivă a scriptului. În fiecare caz, rutina script de tratare a evenimentului este numită `RunMe()`. Exemplul identifică în ce condiții va rula scriptul. Când evenimentul se declanșează, agentul de utilizator rulează funcția script `RunMe()` pe care a întâlnit-o când fișierul HTML a fost analizat inițial.

Ca un prim exemplu, considerați un examen online cu variante de răspuns multiple transmis utilizatorului prin pagini Web HTML 4. Fiecare întrebare are patru răspunsuri posibile, arătate ca imagini bitmap. Când un utilizator execută clic pe un răspuns, imaginea se modifică într-una care identifică un răspuns corect sau unul incorect. Evenimentele ar putea fi configurate după cum urmează:

```
<DIV><IMG ID=Answer1 SRC="Answer1.gif" onClick="RunMe();"></DIV>
<DIV><IMG ID=Answer2 SRC="Answer2.gif" onClick="RunMe();"></DIV>
<DIV><IMG ID=Answer3 SRC="Answer3.gif" onClick="RunMe();"></DIV>
<DIV><IMG ID=Answer4 SRC="Answer4.gif" onClick="RunMe();"></DIV>
```

Funcția script `RunMe()` determină ce răspuns a selectat utilizatorul cu dispozitivul de indicare. Apoi, funcția compară răspunsul selectat cu răspunsul corect. Dacă utilizatorul a selectat răspunsul greșit, valoarea atributului `SRC` este schimbată în `wrong.gif`.

Funcția `RunMe()` se execută când se declanșează evenimentul `onclick`. Amintiți-vă, evenimentul `onclick` se declanșează ori de câte ori un utilizator execută clic pe un element cu butonul unui dispozitiv de indicare.

Pentru cel de al doilea exemplu, am folosit o hartă a Europei de Est care permite utilizatorului să deplaseze dispozitivul de indicare peste o țară pentru a afla mai multe despre geografia sa. Fiecare țară poate fi configurată cu etichetele de diviziune potrivite, după cum urmează:

```
<DIV><IMG ID=Poland SRC="Poland.gif" onMouseOver="RunMe();"></DIV>
```

Funcția script `RunMe()` determină în interiorul cărei țări s-a deplasat dispozitivul de indicare. Când utilizatorul a selectat țara, scriptul transmite un șir lung de text unui paragraf numit în eticheta sa de paragraf și înlocuiește conținutul curent al paragrafului cu text explicativ despre țara selectată. Modificarea conținutului unei etichete fără o cerere suplimentară către un server Web este numită *înlocuire dinamică de conținut HTML (dynamic HTML content replacement)*. Cuprinsul paragrafului explicativ se modifică dinamic într-un text care urmărește totdeauna poziția curentă a cursorului dispozitivului de indicare.

Funcția `RunMe()` se execută când se declanșează evenimentul `onmouseover`. Amintiți-vă, evenimentul `onmouseover` se declanșează ori de câte ori un utilizator deplasează dispozitivul de indicare pe un element.

Pentru cel de al treilea exemplu, am folosit o aplicație de așezare a mobilei într-un birou. Utilizatorul deschide o pagină Web și mută imagini cu vederea de sus a mobilei de birou pentru a crea dispoziția în spațiu a încăperii visate. Fiecare piesă de mobilier ar putea fi configurată cu etichetele de diviziune potrivite, după cum urmează:

```
<DIV><IMG ID=Chair SRC="Chair1.gif" onMouseMove="RunMe();"></DIV>
```

Funcția script `RunMe()` determină peste care piesă de mobilier s-a mutat dispozitivul de indicare. Când utilizatorul selectează o piesă de mobilier, scriptul actualizează poziția ei bazându-se pe locul unde utilizatorul deplasează mouse-ul. De asemenea, scriptul actualizează un index de adâncime pentru articolul pe care utilizatorul îl mută, pentru a face ca acea piesă de mobilier să apară totdeauna deasupra tuturor celorlalte.

Funcția `RunMe()` se execută când se declanșează evenimentul `onmousemove`. Amintiți-vă, evenimentul `onmousemove` se declanșează continuu când un utilizator deplasează dispozitivul de indicare peste respectivul element. Acesta este un lucru diferit față de exemplul cu Europa de Est, în care evenimentul `onmouseover` se declanșa doar o singură dată, când mouse-ul intra în aria fizică acoperită de poziția elementului.

OBSERVAȚIE

Aceste trei exemple de pagini dinamice Web folosesc, toate, imagini ca elemente sursă ale evenimentelor, dar elementele text sunt elemente sursă la fel de valabile. Exemplul inline pe care l-ați văzut mai înainte folosește un element text, un titlu, ca element sursă. Aceste trei exemple ar putea, la fel de ușor, să folosească titluri sau oricare alt element text, în locul imaginilor.

Rezumat

Acest capitol vă oferă un punct de sprijin pentru crearea paginilor Web cu modificări dinamice: *evenimentele intrinseci*. Producătorii de agenți de utilizator înglobează evenimente intrinseci în aplicațiile lor, astfel că autorii Web se pot bizui pe ele. În ceea ce privește autorul de pagini Web, evenimentele sunt *intrinseci* pentru HTML 4. Autorii creează rutine de tratare a evenimentelor pentru a preciza ce acțiune ar trebui să aibă loc atunci când aplicația agent de utilizator observă că un eveniment a avut loc.

În mod curent, există 18 evenimente specificate în schița de lucru a W3C pentru HTML 4. Un autor identifică evenimentele pe care agentul de utilizator at trebui să le trateze prin identificarea lor drept atribute ale etichetelor HTML potrivite. De exemplu, un eveniment `ondblclick` este activat prin includerea unui atribut `ondblclick` în interiorul etichetei elementului care recepționează un clic dublu, iar un eveniment `onload` este activat prin includerea unui atribut `onload` în interiorul etichetei de deschidere `<DIV>` a documentului HTML 4. De cele mai multe ori plasarea atributului de eveniment este ușor de înțeles, dar amintiți-vă că nu toate atributele de eveniment sunt valabile în toate etichetele HTML 4. Acest capitol v-a oferit o explicație concisă a validității și utilizării pentru fiecare eveniment HTML 4 din secțiunea „Lista evenimentelor HTML 4".

O dată ce evenimentele sunt configurate corect în interiorul structurii de etichete HTML 4 a unei pagini Web, rutinele de tratare a evenimentelor dictează ce acțiune are loc când un eveniment se declanșează. Puteți scrie rutine de tratare a evenimentelor ca funcții script separate, așa cum apar în Capitolul 29, sau vă puteți scrie rutinele de tratare a evenimentelor în însăși interiorul structurii de etichetă HTML 4. În oricare eventualitate, puterea rutinelor de tratare a evenimentelor deschide paginile dumneavoastră Web în fața unor posibilități captivante, dinamice și animate.

Capitolul 29
HTML și scripturi – Combinația DHTML

de Bruce Campbell

În acest capitol

- **HTML dinamic 414**
- **De ce HTML dinamic? 415**
- **Crearea paginilor Web dinamice 415**
- **Exemple de pagini HTML dinamice 418**
- **Etichetele <NOSCRIPT></NOSCRIPT> 432**
- **Scripturi transformate în comentariu 433**

HTML dinamic

În capitolul anterior, ați învățat despre modelul de evenimente HTML 4. HTML 4 specifică 18 evenimente pentru utilizare la crearea paginilor Web dinamice și interactive. Acest capitol vă împlinește cunoștințele, arătându-vă cum să folosiți scripturi pentru a desemna rutinele dumneavoastră de tratare a evenimentelor.

Ați învățat în Capitolul 28, „Lucrul cu utilizatorul – Evenimente", cum evenimentele standardizate reclamă coordonare între producătorul agentului de utilizator și autorul paginii Web HTML 4. O dată ce aplicația agent de utilizator încorporează modelul de evenimente HTML 4, autorul scrie rutine de tratare a evenimentelor pentru a configura acțiunile de pe pagina Web. Apoi, pentru multe evenimente, declanșarea evenimentului depinde de acțiunile surferului Web. De exemplu, evenimentul `onclick` cere ca utilizatorul să apese un buton pe dispozitivul său de indicare pentru ca rutina de tratare `onclick` să ruleze pentru elementul asupra căruia s-a executat clic.

Acest capitol acoperă cerințele tehnice și de creativitate pe care trebuie să le îndepliniți pentru a vă face pagina Web mai sensibilă și interactivă cu auditoriul dumneavoastră din Web. Veți citi despre detaliile tehnice de care trebuie să țineți seama la crearea paginilor Web dinamice în concordanță cu specificația HTML 4, fără să pierdeți din vedere inovațiile suplimentare ale producătorilor de browsere Web. Tehnologia pe care o folosiți pentru a crea pagini Web dinamice, altfel un subset al HTML 4, se conturează distinct, ca o tehnologie independentă, sub numele general acceptat de *HTML dinamic (dynamic HTML)*.

HTML dinamic își merită eticheta, deoarece aduce paginilor Web caracteristici pe care utilizatorii le pretind în alte aplicații curente. Cu toate că Webul a crescut o dată cu HTML 1, 2 și 3.2, care au ajutat autorii de pagini Web să creeze pagini întrucâtva sofisticate, ca de enciclopedie, mulți utilizatori preferă ca paginile Web să se comporte la fel ca alte aplicații pe care le folosesc zilnic.

Cum s-a menționat în capitolul anterior, multe dintre evenimentele HTML 4 ale schiței de lucru preiau avantajul evenimentelor a căror evidență este deja păstrată de sistemul de operare. HTML dinamic comunică acele evenimente paginii Web. Un eveniment devine semnificativ pentru o pagină Web numai dacă dumneavoastră, ca autor de pagină Web, menționați în mod specific efectul său în interiorul documentului dumneavoastră HTML. Acest capitol prezintă exemple complete de funcții script de tratare a evenimentelor, pe care le puteți extinde sau de la care puteți învăța să creați pagini Web care prind esența HTML-ului dinamic.

De ce HTML dinamic?

În afară de a ajuta paginile Web să ajungă la o stare matură pe calculatoarele sofisticate ale zilelor noastre și de a angaja utilizatorii printr-o selectivitate și interactivitate crescută, HTML-ul dinamic adaugă eficiență modului în care Web livrează documente, pe care unii îl consideră deficitar. Documentele Web sunt transmise utilizatorilor Web folosind HTTP (HyperText Transport Protocol, protocol de transport al hipertextelor). Versiunea de HTTP folosită predominant pe Web astăzi cere o suprasarcină (overhead) mare pentru fiecare fișier pe care un server Web îl livrează unui client.

La fel cum poștașul din vecinătate ar fi ineficient dacă s-ar întoarce la oficiul poștal de fiecare dată să ia corespondența pentru următoarea adresă din drum, distribuirea siturilor Web fișier cu fișier poate fi ineficientă. Desigur, standardul HTTP este în curs de transpunere într-o formă nouă, pentru a elimina întrucâtva ineficiența sa, dar HTML-ul dinamic oferă autorilor Web ocazia de a livra mai multe informații într-un singur fișier. De fapt, ați putea transmite echivalentul textului dintr-un întreg sit Web ca pe un singur fișier. Totuși, fișierele dumneavoastră grafice mai sunt, încă, livrate câte unul o dată când folosiți HTML dinamic, așa cum erau cu HTML 3.2.

HTML-ul dinamic oferă beneficii serverului Web și reduce cheltuielile suplimentare de comunicație prin protocol în rețea. Folosind HTML dinamic, serverele Web reclamă mai puține cereri de conectare decât o făceau cu documentele compatibile HTML 3.2. Poate că ați luat în considerare schimbări similare când ați proiectat siturile dumneavoastră Web pentru a avea documente hiperlegate reduse sau câteva documente extinse marcate din plin cu semne de carte. În unele privințe, chestiunile sunt asemănătoare dacă ne referim la compromisurile utilizării HTML-ului dinamic față de HTML 3.2.

Puteți folosi HTML-ul dinamic pentru a trimite un fișier mare de informații o dată unui utilizator, în loc de multe pagini Web mai mici. HTML 3.2 reclamă o planificare similară. Marele beneficiu nou cu HTML dinamic este totuși acela că utilizatorul nu trebuie să fie inundat cu informațiile întregii pagini încă de la început. Puteți prezenta informațiile treptat, ca răspuns la acțiunile unui utilizator. Acțiunile rezultă din rutinele dumneavoastră de tratare a evenimentelor.

Este de ajuns să spun că HTML-ul dinamic implică alte strategii și compromisuri în distribuirea paginii Web decât în cazul versiunii HTML 3.2. Principalul scop al acestui capitol este să vă ajute să folosiți HTML-ul dinamic în mod corect și creativ. Chiar dacă nu apreciați oportunitățile noi pentru sarcini reduse în comunicații și conectarea serverului Web, vă veți bucura să treziți mai ușor la viață paginile dumneavoastră Web.

Crearea paginilor Web dinamice

Creați pagini Web dinamice adăugând scripturi de tratare a evenimentelor la documentele dumneavoastră HTML. Amintiți-vă că scripturile dumneavoastră de tratare a evenimentelor devin active când un eveniment este declanșat printr-un element din pagina Web. Restul acestui capitol oferă exemple complete de pagini Web dinamice. Când stăpâniți fiecare exemplu, jucați-vă în mod creativ cu codul pentru a schimba evenimente, adăuga noi funcții și combina efecte.

Scripturi de tratare a evenimentelor

Scrieți scripturi de tratare a evenimentelor cu un limbaj de scriptare potrivit. Specificația HTML nu spune nimic despre standardizarea limbajului de scriptare în domeniul său. În schimb, cei mai mulți autori de limbaje de scriptare care obișnuiesc să creeze pagini Web dinamice și-au format discipoli în ultimii câțiva ani. Scripturile incluse în paginile Web HTML 2 sau HTML 3.2 se execută atunci când pagina Web se încarcă inițial. Printre limbajele populare de scriptare a paginii Web se numără JavaScript, VBScript și TCL. W3C oferă exemple de rutine script de tratare a evenimentelor pentru toate trei limbajele.

Includerea scripturilor într-un document HTML

Cu excepția cazului în care nu folosiți decât simpla tratare inline a evenimentelor, așa cum s-a discutat în Capitolul 28, veți plasa rutinele dumneavoastră de tratare a evenimentelor în interiorul unui element script. Elementul `SCRIPT` constă dintr-o pereche de etichete, `<SCRIPT></SCRIPT>`, atributele corespunzătoare etichetei de deschidere `<SCRIPT>` și una sau mai multe funcții script plasate între eticheta de deschidere și cea de închidere. Elementul `SCRIPT` poate să apară de ori câte ori în interiorul elementului `HEAD` sau al elementului `BODY` al unui document HTML.

Dispuneți și de opțiunea de a furniza funcțiile script dintr-un fișier extern și de a folosi atributul `src` al elementului `SCRIPT` pentru a face trimiteri la ele. Pentru agenții de utilizator care nu tratează sau nu pot trata scripturi, autorii pot include conținut alternativ folosind elementul `NOSCRIPT`.

O etichetă de deschidere `<SCRIPT>` conține, în mod obișnuit, unul sau mai multe dintre cele trei atribute: `type`, `language` și `src`. Atributul `type` precizează tipul de conținut pentru limbajul de scriptare a cărui sintaxă este folosită în funcția script din elementul script. Valoarea atributului `type` trebuie să fie un tip de medii Internet (Internet Media Type). Nu există nici o valoare prestabilită pentru atributul `type`. Atributul `language` reprezintă un alt mod de a identifica un limbaj de scriptare, dar W3C preferă să folosiți, în loc, atributul `type`. Atributul `src` specifică locația unui script extern. Valoarea pentru atributul `src` poate fi o adresă URL relativă sau absolută. Dacă atributul `src` are o valoare URL, agenții de utilizator trebuie să ignore cuprinsul elementului și să regăsească scriptul folosind adresa URL.

Limbajul de scriptare trebuie să fie identificat explicit pe fiecare pagină Web pentru a fi compatibil HTML 4. Dacă prevedeți funcții script multiple într-un singur fișier HTML, vă puteți identifica limbajul de scriptare o singură dată, prin plasarea etichetei `<META>` corespunzătoare în interiorul perechii de etichete `<HEAD></HEAD>`. De exemplu, puteți defini JavaScript ca limbajul dumneavoastră de scriptare prin elementul `<HEAD>`, așa cum se arată în continuare:

```
<HEAD>
<META http-equiv="Content-Script-Type" content="text/javascript">
</HEAD>
```

Aici, valoarea atributului `content` este un tip valabil de medii Internet.

W3C numește un limbaj de scriptare definit printr-o etichetă `<META>` *limbaj de scriptare prestabilit (default scripting language)*, deoarece sintaxa respectivului limbaj de scriptare este așteptată de agentul de utilizare în lipsa oricărui alt atribut `type` din etichetele de deschidere `<SCRIPT>`. Puteți folosi limbaje de scriptare multiple pe aceeași pagină Web câtă vreme încadrați rutinele script în interiorul perechii de etichete `<SCRIPT></SCRIPT>` corespunzătoare; fiecare pereche identifică limbajul ei de scriptare. Există două moduri prin care puteți identifica un limbaj de scriptare pentru un element script. Puteți folosi atributul `type` al elementului de scriptare:


```
<SCRIPT type="text/javascript">
```

Sau puteți folosi atributul `language` al elementului script:

```
<SCRIPT language=javascript>
```

Totuși, W3C recomandă să folosiți atributul `type` pentru a fi asemănător specificației etichetei `<META>`. Declararea locală a limbajului de scriptare se suprapune efectiv peste declararea prestabilită, dacă există. Amintiți-vă că documentele HTML 4 care nu conțin nici o declarație de limbaj de scriptare prestabilit, nici una locală pentru un element script sunt incorecte. Agenții de utilizator ar trebui să încerce să interpreteze scriptul, dar nu li se cere să o facă.

Fiecare limbaj de scriptare are propriile sale reguli de referire la obiectele HTML din interiorul unui script. Specificația HTML 4 nu definește un mecanism standard de referire la obiectele HTML, însă scripturile trebuie să se refere la un element conform cu numele său atribuit. Motoarele de scriptare ar trebui să considere următoarea regulă de precedență la identificarea unui element: un atribut `name` are întâietate asupra unuia `id`, dacă ambele sunt setate. Altminteri, pot fi folosite ori unul, ori celălalt. Conținutul elementului `SCRIPT` este un script, așa încât el nu trebuie să fie evaluat de către agentul de utilizator ca marcaj HTML. Agentul de utilizator trebuie să îl transmită mai departe ca date unui motor de scriptare.

FOLOSIREA ETICHETELOR DE MARCARE HTML ÎN SCRIPTURI

Dacă scrieți scripturi sofisticate, ați putea dori să faceți trimiteri la etichete de marcare HTML în codurile dumneavoastră de script. W3C este foarte exact referitor la modul în care ar trebui să definească un agent de utilizator unde începe un script și unde sfârșește în interiorul elementului script. Înștiințarea explicativă W3C este reprodusă mai jos:

„Analizoarele HTML trebuie să fie capabile să recunoască datele script ca începând imediat după eticheta de start și sfârșind îndată ce delimitatorii `ETAGO` (`</`) sunt urmați de un caracter nume ([a-zA-Z]). Datele script nu se termină, în mod necesar, cu eticheta de sfârșit `</SCRIPT>`, dar sunt încheiate prin orice `</` urmat de un caracter nume.

Prin urmare, oricare marcaj HTML care se intenționează a fi trimis unui motor de scriptare (care poate să facă orice dorește cu marcajul) trebuie să fie „codat escape" pentru a nu duce în eroare analizorul HTML. Proiectanții oricărui limbaj de scriptare ar trebui să recomande suportul specific de limbaj pentru rezolvarea acestei chestiuni."

Iată un exemplu ilegal:

Codul care urmează este invalid datorită prezenței caracterelor `</EM>` în elementul `SCRIPT`:

```
<SCRIPT type="text/javascript">
   document.write ("<EM>This won't work</EM>")
</SCRIPT>
```

Un analizor conform specificației trebuie să trateze datele `</EM>` ca sfârșit al datelor scriptului, ceea ce este clar altceva decât intenționa autorul.

În JavaScript, acest cod poate fi exprimat legal asigurând ca delimitatorul vizibil `ETAGO` să nu apară imediat înaintea unui caracter de început al unui nume SGML:

```
<SCRIPT type="text/javascript">
   document.write ("<EM>This will work<\/EM>")
</SCRIPT>
```

În TCL, puteți scrie aceasta ca un cod legal astfel:

```
<SCRIPT type="text/tcl">
   document write "<EM>This will work<\/EM>"
</SCRIPT>
```

În VBScript, problema poate fi evitată cu funcția `Chr()`:

```
"<EM>This will work<\" & Chr(47) + "EM>"
```

Scripturile pot fi utilizate cu documente HTML 4 pentru a produce pagini Web cu mare interactivitate. W3C încadrează caracteristicile potențiale activate de script în următoarele patru categorii:

1. Scripturi care sunt evaluate în timp ce se încarcă documentul pentru a-i modifica cuprinsul în mod dinamic.

2. Scripturi care însoțesc un formular pentru a prelucra datele de intrare când sunt introduse. Proiectanții pot să completeze în mod dinamic părți ale unui formular, bazându-se pe valorile din alte câmpuri. Ei se pot și asigura că datele de intrare se conformează șirurilor predeterminate de valori, că respectivele câmpuri sunt reciproc compatibile ș.a.m.d..

3. Scripturi care sunt declanșate de evenimente care afectează documentul, ca încărcarea, descărcarea, focalizarea elementului, mișcarea mouse-ului ș.a.m.d.

4. Scripturi care sunt legate la controale de formular (cum ar fi butoanele) pentru a produce elemente grafice ale interfeței de utilizator.

Acest capitol continuă cu exemple de rutine de tratare a evenimentelor pentru fiecare dintre cele patru categorii anterioare. Dacă citiți fiecare exemplu, gândiți-vă în mod creator și abstract despre cum ați putea folosi sau extinde exemplul pentru a aduce noi trăsături paginilor dumneavoastră Web HTML 4.

Exemple de pagini HTML dinamice

Paginile Web dinamice constau din etichete HTML, text de rând, atribute de eveniment și rutine de tratare a evenimentului. Etichetele și atributele HTML s-au dezvoltat din modestele lor începuturi din specificația HTML 1. Atributele de eveniment sunt o caracteristică nouă în HTML 4, asociată cu lista de evenimente intrinseci tratată în Capitolul 28. Rutinele-script de tratare sunt incluse în interiorul perechii de etichete `<SCRIPT></SCRIPT>`. Oricare alt text dintr-un fișier HTML care nu este între simbolurile < și > ale fiecărei etichete HTML sau în interiorul unui element script este considerat text de rând.

Veți observa toate patru componentele (etichete HTML, textul de rând, atribute de eveniment și rutine-script de tratare) în fiecare exemplu din această secțiune. Familiarizați-vă cu ele, identificând fiecare componentă. La sfârșit, luați în considerare moduri de a combina componentele din fiecare exemplu în feluri noi și captivante. Cele cinci exemple care urmează reprezintă cele patru tipuri de categorii din secțiunea precedentă: un exemplu de dinamică la momentul încărcării, un exemplu de prelucrare de formular, două exemple de acțiune a utilizatorului și un exemplu de interfață cu utilizatorul.

Un exemplu de dinamică la momentul încărcării

Pagini Web dinamice au devenit disponibile de ceva vreme autorilor care scriu pentru Netscape Navigator și Internet Explorer. În browserele anterioare celor compatibile HTML, toate scripturile sunt încărcate la momentul încărcării și completate înainte ca pagina Web să-și încheie încărcarea. Exemple sunt paginile Web care accesează servere Web dinamice pentru a determina o valoare modificată sub controlul serverului. De exemplu, paginile Web pot accesa un server dedicat bursei, un server dedicat timpului probabil, un server dedicat jocurilor sau un server de știri pentru a obține informații care se schimbă în mod regulat. Pagina Web actualizează o variabilă din script și scrie valoarea corespunzătoare. Valoarea devine text static până la viitoarea dată când pagina este încărcată. Un server Web care actualizează continuu o pagină Web din inițiativa sa folosește ceea ce se cheamă *tehnologie push (push technology)*. Tehnologia push așază greul poverii pe server, dar calculatorul casnic tipic zilelor noastre este echipat cu putere de procesare semnificativă.

HTML dinamic preia avantajul puterii de procesare a calculatorului client. Un exemplu semnificativ de pagină Web cu dinamică în timpul încărcării este o pagină Web animată care începe automat la momentul încărcării. Exemplul din Listingul 29.1 reprezintă o pagină Web care rulează o animație a procesului de urmărire prin sonar, folosit de biologii marini la urmărirea formelor de viață acvatice.

Exemplul rulează sub Internet Explorer 4.0, folosind controale ActiveX care se încarcă la o instalare standard. Toate etichetele HTML ar trebui să vă fie familiare dacă ați citit capitolele anterioare care acoperă specificația HTML 4. Dați atenție specială atributului de eveniment `onload` și scriptului de tratare a evenimentului `cycle()`. Aruncați o privire Listingului 29.1 pentru a vedea sintaxa de fișier HTML și Figurii 29.1 pentru a vedea fișierul încărcat într-un browser Web.

Listingul 29.1. O animație la momentul încărcării.

```
<HTML>
<HEAD>
</HEAD>
<BODY BACKGROUND="water.jpg" onload="cycle();"
 TEXT=#FFFFFF LINK=FFFF22 VLINK=FF22FF>
<H2>Squid Tracking -- How Do They Do It?</H2>
<TABLE>
<TR><TD WIDTH=320>
<DIV ID=AnimDiv STYLE="position:relative;width:320px;height:320px">
<IMG SRC="sonard.gif" ID=sonard BORDER=0
 STYLE="container:positioned;position:absolute;
 TOP:0pt;LEFT:150px;WIDTH:32px;HEIGHT:32px;ZINDEX:0;">
<IMG SRC="sonaru.gif" ID=sonaru BORDER=0
 STYLE="container:positioned;position:absolute;
 TOP:0pt;LEFT:150px;WIDTH:32px;HEIGHT:32px;ZINDEX:0;">
<IMG SRC="squid2.gif" ID=squid1 BORDER=0 onclick="Guess();"
 STYLE="container:positioned;position:absolute;
 TOP:170pt;LEFT:20px;WIDTH:64px;HEIGHT:64px;ZINDEX:1;">
<IMG SRC="squid2.gif" ID=squid2 BORDER=0 onclick="Guess();"
 STYLE="container:positioned;position:absolute;
 TOP:190pt;LEFT:20px;WIDTH:64px;HEIGHT:64px;ZINDEX:2;">
<IMG SRC="squid2.gif" ID=squid3 BORDER=0 onclick="Guess();"
 STYLE="container:positioned;position:absolute;
 TOP:180pt;LEFT:0px;WIDTH:64px;HEIGHT:64px;ZINDEX:3;">
<IMG SRC="boat.jpg" BORDER=0 STYLE="container:positioned;position:
 absolute; TOP:0pt;LEFT:0px;WIDTH:320px;HEIGHT:50px;ZINDEX:5;">
</DIV></TD><TD>
<FONT COLOR=#FFFF00><I>To track squid, they use a sophisticated sonar
device that emits sound waves down into the ocean from the surface.
The sound waves will either bounce off the ocean floor or bounce off
the squid if they get between the emitting boat and the ocean floor.
In this manner, they get very precise images of each squid that gets in the way
of the sound waves.</I></FONT><BR>
</TD></TR></TABLE>
<OBJECT ID="pathone"
     CLASSID="CLSID:E0E3CC60-6A80-11D0-9B40-00A0C903AA7F">
<PARAM NAME=XSeries
     VALUE="0,20;10,50;20,80;30,115;40,150;50,185;60,220;70,255;80,300">
```

(continuare)

Listingul 29.1. - Continuare

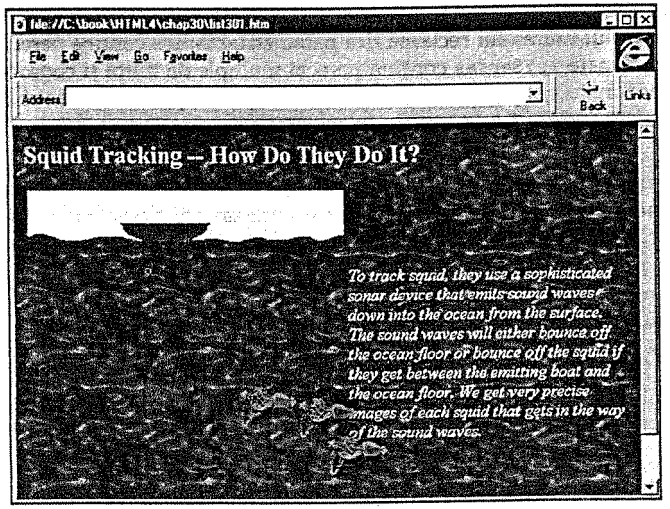
```
<PARAM NAME=YSeries
      VALUE="0,170;10,170;20,170;30,170;40,170;50,170;60,170;70,170;80,170">
</OBJECT>
<OBJECT ID="pathtwo"
      CLASSID="CLSID:E0E3CC60-6A80-11D0-9B40-00A0C903AA7F">
<PARAM NAME=XSeries
      VALUE="0,20;10,55;20,80;30,110;40,135;50,170;60,200;70,230;80,270">
<PARAM NAME=YSeries
      VALUE="0,190;10,190;20,193;30,195;40,197;50,194;60,192;70,190;80,190">
</OBJECT>
<OBJECT ID="paththree"
      CLASSID="CLSID:E0E3CC60-6A80-11D0-9B40-00A0C903AA7F">
<PARAM NAME=XSeries
      VALUE="0,0;10,20;20,50;30,80;40,100;50,128;60,148;70,178;80,210">
<PARAM NAME=YSeries
      VALUE="0,180;10,182;20,185;30,190;40,185;50,181;60,179;70,176;80,179">
</OBJECT>
<OBJECT ID="pathfour"
      CLASSID="CLSID:E0E3CC60-6A80-11D0-9B40-00A0C903AA7F">
<PARAM NAME=XSeries
      VALUE="0,150;10,150;20,150;30,150;40,150;50,150;60,150;70,150;80,150">
<PARAM NAME=YSeries
      VALUE="0,20;10,50;20,90;30,135;40,180;41,0">
</OBJECT>
<OBJECT ID="pathfive"
      CLASSID="CLSID:E0E3CC60-6A80-11D0-9B40-00A0C903AA7F">
<PARAM NAME=XSeries
      VALUE="0,150;10,150;20,150;30,150;40,150;50,150;60,150;70,150;80,150">
<PARAM NAME=YSeries
      VALUE="40,0;41,180;50,135;60,90;70,50;80,20">
</OBJECT>
<SCRIPT language="vbscript">
function cycle()
        dim itimer
        pathone.Target = squid1.Style
        pathtwo.Target = squid2.Style
        paththree.Target = squid3.Style
        pathfour.Target = sonard.Style
        pathfive.Target = sonaru.Style
        squid1.style.visibility="visible"
        squid2.style.visibility="visible"
        squid3.style.visibility="visible"
        sonard.style.visibility="visible"
        sonaru.style.visibility="visible"
        pathone.Play
        pathtwo.Play
        paththree.Play
        pathfour.Play
        pathfive.Play
        itimer = setTimeout( "Moves()", 50)
End function

Sub Moves
        dim itimer
```



```
        pathone.Tick
        pathtwo.Tick
        paththree.Tick
        pathfour.Tick
        pathfive.Tick
        if currentTick < 9 then
            itimer = setTimeout( "Moves()", 50)
        end if
End Sub
</SCRIPT>
</BODY>
</HTML>
```

Figura 29.1 arată animația de urmărire prin sonar a sepiilor pe măsură ce undele sonarului se întorc la suprafața oceanului. În momentul prezentat din secvența de animație, undele sonarului au fost reflectate deja de sepiile în mișcare.

FIGURA 29.1. *O animație în momentul încărcării.*



Eticheta de deschidere `<BODY>` include singurul atribut de eveniment din acest exemplu. Pentru a rula un script imediat după ce pagina Web se încarcă, plasați un atribut `onload` în eticheta `<BODY>` și furnizați un nume de funcție script ca valoare a sa. Numai evenimentul `onload` poate fi pus în eticheta de deschidere `<BODY>`. În Listingul 29.1, numele funcției script de animație este `cycle()`, care este scrisă folosind limbajul de scriptare VBScript. Browserul Web procesează corect scriptul deoarece eticheta de deschidere `<SCRIPT>` conține atributul necesar `language="vbscript"`.

OBSERVAȚIE

Toate exemplele din acest capitol și din cel următor au fost testate folosind browserul Web Internet Explorer 4.0. Deoarece specificația HTML 4 se schimbă de-a lungul timpului, puteți continua să folosiți aceste exemple pentru a obține o înțelegere generală a ceea ce oferă HTML-ul dinamic autorilor de pagini Web. Vizitați Webul, în special siturile Web ale Netscape (`www.netscape.com`), Microsoft (`www.microsoft.com`) și W3C (`www.w3c.org`), pentru orice modificări de sintaxă, model de obiecte sau model de evenimente.

Funcția `cycle()` apelează o subfuncție numită `Moves`. În funcția `cycle()`, fiecare imagine de sepie sau de sonar este asociată cu un obiect de sincronizare prin intermediul unei linii ca aceasta:

```
pathone.Target = squid1.Style.
```

Fiecare obiect sepie sau sonar posedă o metodă `Play` care este apelată pentru a porni secvența de sincronizare. Metoda `Play` este activată pentru fiecare imagine de sepie și sonar cu o linie de felul acesteia:

```
pathone.Play
```

De asemenea, obiectele de sincronizare posedă o metodă `Tick` care incrementează indexul în intervale de sincronizare. Metoda `Tick` este apelată în subfuncția `Moves`, care asigură construcția în buclă pe care o cere animația. Secvența de sincronizare și țintele pentru cale sunt comunicate browserului Web prin obiecte ActiveX. Obiectele ActiveX sunt încărcate prin perechile de etichete `<OBJECT></OBJECT>`.

Fiecare etichetă `<OBJECT>` conține un atribut `CLASSID="CLSID:E0E3CC60-6A80-11D0-9B40-00A0C903AA7F">`, care identifică pentru browserul Web controlul ActiveX corespunzător. Controlul animației reclamă doi parametri. `XSeries` conține perechi multiple de timpi și coordonate x; `YSeries` conține perechi multiple de timpi și coordonate y. Combinația de coordonate x și y definește secvența de animație pentru fiecare obiect imagine animat.

În pagina Web creată de Listingul 29.1, animația începe de îndată ce pagina se încarcă. O dată animația terminată, utilizatorul are nevoie să reîncarece pagina pentru a reveni la animație. Ați putea adăuga imaginii un atribut `onclick` cu atributul `SRC="boat.jpg"`, pentru a porni animația ori de câte ori utilizatorul a executat clic pe barca de la suprafața apei. Scriptul de tratare a evenimentului rămâne exact același și tratează atât evenimentul `onload`, cât și evenimentul `onclick`.

Un exemplu de prelucrare de formular

Pe măsură ce Webul își împlinește promisiunile de tehnologie care acceptă informații de brokeraj și tranzacții comerciale, tot mai multe formulare electronice apar pe paginile Web pentru a solicita informații de la surferii Web. HTML-ul dinamic ne ajută să creăm formulare cu interactivitate crescută, furnizând evenimente cum ar fi `onblur` și `onchange`, pe care un autor de pagină Web le poate utiliza să verifice și să valideze introducerile în câmpuri de formulare.

Listingul 29.2 este un exemplu simplu de prelucrare de formulare folosind HTML dinamic și evenimentul intrinsec `onchange`. Cele două câmpuri de date incluse în pagina Web din Listingul 29.2 creează o tranzacție de comandă a unor bilete de concert în care un utilizator poate comanda numai 10 bilete o dată. Formularul calculează totalul comenzilor respective, bazându-se pe cantitatea introdusă, atâta vreme cât aceasta se păstrează la 10 sau mai puțin. Fiecare bilet costă 8,88 USD și nu se aplică comision de vânzare. Priviți Listingul 29.2 și Figura 29.2, iar apoi continuați să citiți pentru mai multe explicații asupra acestui exemplu.

Listingul 29.2. Un exemplu de prelucrare de formular.

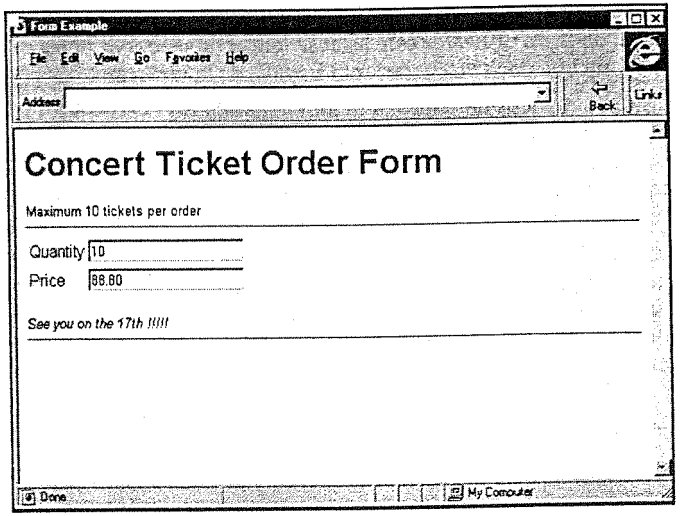
```
<HTML>
<HEAD>
<FONT FACE-"verdana,arial,helvetica" SIZE=2>
<TITLE>Form Example</TITLE>
</HEAD>
<BODY BGCOLOR="#FFFFFF">
<H1>Concert Ticket Order Form</H1>
```



```
Maximum 10 tickets per order
<HR>
<TABLE>
<TR><TD>Quantity</TD>
<TD><INPUT NAME="qty" onChange="checkNum();" VALUE="0">
</TD></TR>
<TR><TD>Price</TD>
<TD><INPUT NAME="price" VALUE="">
</TD></TR>
</TABLE>
<P><I>See you on the 17th !!!!!</I>
<SCRIPT language=javascript>
function checkNum() {
  if (qty.value >= 10) {
     qty.value=10;
  }
  price.value=qty.value*8.88;
}
</SCRIPT>
<HR>
</BODY>
</HTML>
```

Figura 29.2 arată formularul de comandă a biletelor de concert după ce un utilizator încearcă să introducă o cantitate de 15 bilete. Formularul acceptă o cantitate maximă de numai 10 bilete, drept care datele introduse de utilizator sunt schimbate la 10 și este calculat prețul de 88,80 USD pentru prețul total al comenzii.

FIGURA 29.2. *Un exemplu de prelucrare de formular.*



Exemplul din Listingul 29.2 este elementar comparat cu exemplul de animație din Listingul 29.1. Țineți minte că paginile Web dinamice pot fi tot atât de complicate sau de simple pe cât dictează nevoia dumneavoastră. Interactivitatea este asigurată de atributul `onchange`, care este parte a etichetei `<INPUT>` cu atributele `NAME="Qty"` și `VALUE="0"`. Atributul `onchange`

are valoarea checkNum(), care identifică o funcție script de tratare a evenimentului pentru evenimentul onchange.

OBSERVAȚIE

Restul exemplelor din acest capitol folosesc, toate, JavaScript ca limbaj de scriptare. Înainte să apară această carte, JavaScript era, de departe, cel mai răspândit limbaj de scriptare în rândul dezvoltatorilor de aplicații Web care scriu pentru agenți de utilizator multipli.

Funcția checkNum() este înglobată în perechea de etichete <SCRIPT></SCRIPT>. Eticheta de deschidere <SCRIPT> specifică limbajul de scriptare cu atributul language=javascript. Funcția checkNum() determină valoarea curentă a câmpului text numit Qty și o compară cu maximul de 10. Când cantitatea introdusă este mai mare ca 10, scriptul stabilește valoarea la exact 10. Scriptul calculează, apoi, un preț total pentru comandă și stabilește valoarea câmpului text Field la cantitatea de bilete comandată înmulțită cu 8,88 USD.

Scriptul ar putea, de asemenea, expedia informațiile asupra comenzii către procesul corespunzător pentru a trimite biletele utilizatorului prin poștă, dacă utilizatorul este conectat cu un număr de cont. Formularele bazate pe HTML 4 nu mai sunt dependente exclusiv de un buton de transmitere, dar țineți cont că utilizatorii dumneavoastră s-ar putea aștepta la confirmarea introducerii lor de date înainte de procesare.

Două exemple de acțiune a utilizatorului

Ideea de bază a HTML-ului dinamic este capacitatea vizitatorilor de a se juca cu paginile dumneavoastră Web. Modelul de evenimente vă permite să distribuiți pagini pe care utilizatorii execută clic, peste care trec cu dispozitivul lor de indicare sau cărora chiar le vorbesc, dacă posedă tehnologia de activare vocală adecvată. Rutinele dumneavoastră care tratează evenimentele pot fi foarte sensibile la curiozitatea indiscretă a auditoriului. De fapt, dacă un utilizator este lăsat să interacționeze, s-ar putea ca pagina Web să nu mai semene nici pe departe cu varianta inițială, de la încărcare.

Scripturile de acțiune a utilizatorului sunt atât de captivante, încât această secțiune are două exemple. Primul extinde și restrânge știri pentru diferite orașe dintr-o vale rurală. Al doilea oferă un model pentru distribuția unor informații oarecare prin Web. Primul exemplu este activat de evenimentul mouseover, iar al doilea exemplu vă răspunde la evenimentul onclick.

Listingul 29.3 vă arată fișierul HTML al unei pagini Web de știri pentru o vale rurală. Pagina Web se încarcă cu toate cele șase nume de orașe vizibile, împreună cu temperatura maximă prognozată pentru acea zi. Dacă un utilizator trece cursorul dispozitivului său de indicare peste un nume de oraș, știrile de vârf ale zilei pentru respectivul oraș se extind la dreapta numelui orașului. Figurile 29.3 și 29.4 arată Listingul 29.3 încărcat într-un browser Web.

Listingul 29.3. Un exemplu de extindere și restrângere.

```
<HTML>
<BODY BGCOLOR="#FFFFFF">
<HEAD>
<FONT FACE-"verdana,arial,helvetica" SIZE=3>
<TITLE>Today's High Temperature</TITLE>
</HEAD>
<BODY>
<STYLE>
```



```
.redPlain    {color:rgb(255,0,0);
              font-size:10pt;
              font-style:normal;}
.redSmall    {color:rgb(255,0,0);
              font-size:2pt;
              font-style:normal;}
.blackPlain  {color:rgb(0,0,0);
              font-style:normal;
              font-size:16pt;}
</STYLE>
<H3>The Verentian Valley News</H3>
<HR>
<TABLE>
<TR>
<TD WIDTH=200>
<P ID=T1 CLASS="redPlain" onmouseover="redP();">Argunk     71
<P ID=T2 CLASS="redPlain" onmouseover="redP();">Henders    63
<P ID=T3 CLASS="redPlain" onmouseover="redP();">Krikett    69
<P ID=T4 CLASS="redPlain" onmouseover="redP();">Lowville   68
<P ID=T5 CLASS="redPlain" onmouseover="redP();">Viennes    75
<P ID=T6 CLASS="redPlain" onmouseover="redP();">Yinsburgh  75
</TD>
<TD>
<P ID=PG1 CLASS="redSmall">
The mayor here, Cora James, announces a new plan for curtailing the
current trend of juvenile delinquency: an evening curfew at 9PM.
<P ID=PG2 CLASS="redSmall">
500 community residents continue to keep a lid on the details of
the theme for this year's Solstice Day Festival yet smile smugly
when asked about it.
<P ID=PG3 CLASS="redSmall">
This year farmers suggest a winter wheat crop may be a good idea.
It has been 6 years since the last attempt.
<P ID=PG4 CLASS="redSmall">
Once again, the boy wonder, Jimmy Westen, has made statewide news
for his latest invention.
<P ID=PG5 CLASS="redSmall">
Our regional professional baseball squad wins its fifth straight
and take solid hold of second place.
<P ID=PG6 CLASS="redSmall">
Actress Jennifer Urgent returns home to find community support before
considering her next effort.
</TD>
</TR>
</TABLE>

<SCRIPT LANGUAGE=JavaScript>
var source
function redP() {
  source = window.event.srcElement;
  T1.className="redPlain";
  T2.className="redPlain";
  T3.className="redPlain";
  T4.className="redPlain";
```

(continuare)

Listingul 29.3. – Continuare

```
  T5.className="redPlain";
  T6.className="redPlain";
  PG1.className="redSmall";
  PG2.className="redSmall";
  PG3.className="redSmall";
  PG4.className="redSmall";
  PG5.className="redSmall";
  PG6.className="redSmall";
  source.className="blackPlain";
  if (T1.className="blackPlain")
      PG1.className="blackPlain";
  if (T2.className="blackPlain")
      PG2.className="blackPlain";
  if (T3.className="blackPlain")
      PG3.className="blackPlain";
  if (T4.className="blackPlain")
      PG4.className="blackPlain";
  if (T5.className="blackPlain")
      PG5.className="blackPlain";
  if (T6.className="blackPlain")
      PG6.className="blackPlain";
}
</SCRIPT>
</FONT>
</BODY>
</HTML>
```

Figura 29.3 arată pagina Web de știri din valea rurală imediat după ce a fost încărcată într-un browser Web. În acest moment, toate știrile sunt restrânse.

FIGURA 29.3. *Înainte de declanșarea evenimentului* `mouseover`.

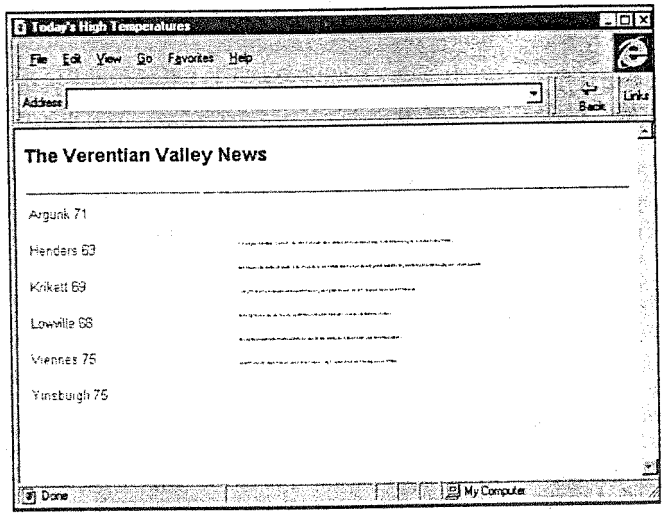
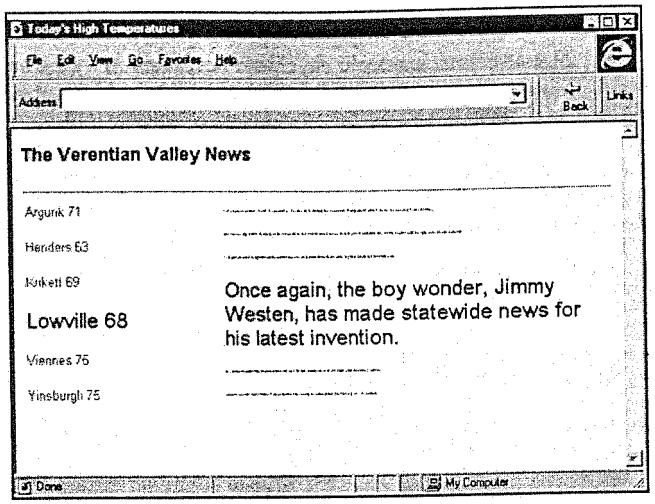


Figura 29.4 arată pagina cu știri din valea rurală imediat după ce un utilizator mută cursorul dispozitivului său de indicare peste textul `Lowville 68`. În acest moment, știrea despre Jimmy Westen se extinde, astfel că poate fi citită ușor.

FIGURA 29.4. *După declanșarea evenimentului* `onmouseover`.



Pagina Web dinamică de știri din valea rurală preia avantajul specificației CSS (foi de stil în cascadă), care este parte din HTML 4. Trei stiluri sunt configurate în perechea de etichete `<STYLE></STYLE>`. Stilul de clasă `.redPlain` este un stil cu text roșu, de 10 puncte tipografice, stilul de clasă `.redSmall` este un stil cu text roșu, de 2 puncte tipografice, iar stilul de clasă `.blackPlain` este un stil de text negru, de 16 puncte tipografice.

Primele șase etichete `<P>`, `T1` până la `T6`, se încarcă, inițial, cu stilul de clasă `.redPlain`. Fiecare paragraf conține un nume de oraș și temperatura maximă a zilei. Toate cele șase etichete `<P>` includ atributul evenimentului `onmouseover` cu o valoare `redP()`. Funcția `redP()` este numele funcției script de tratare a evenimentului pentru evenimentul `onmouseover`.

Următoarele șase etichete `<P>`, numite `PG1` până la `PG6`, se încarcă, inițial, cu stilul de clasă `.redSmall`. Fiecare paragraf conține o știre de top.

OBSERVAȚIE

Ați observat, probabil, inconsecvențe în folosirea tipului de literă (mare sau mică) în exemplele din acest capitol. Doresc să arăt diferite utilizări ale tipului de litere când tehnologia nu este sensibilă la litere mari sau mici. În general, JavaScript este mai sensibil la tipul literelor decât VBScript. De aceea, orice variație a tipului din aceste exemple, va fi acceptată și în exemple VBScript.

Funcția `redP()` este încadrată de o pereche de etichete `<SCRIPT></SCRIPT>`. Din nou, Listingul 29.3 folosește limbajul de scriptare `javascript`, identificat ca un atribut al etichetei de deschidere `<SCRIPT>`. Scriptul `redP()` utilizează o caracteristică foarte utilă a modelului de obiect Internet Explorer. După declararea unei variabile numită `source`, scriptul atribuie sursa evenimentului `mouseover` variabilei sursă:

```
source = window.event.srcElement;
```

Această linie precizează peste care nume de oraș utilizatorul trece cursorul dispozitivului de indicare în acel moment.

Restul scriptului stabilește stilul de text corespunzător fiecărui paragraf de pe pagina Web, bazându-se pe poziția curentă a cursorului dispozitivului de indicare. Dacă ați fi scriptat pentru un browser care nu include proprietatea `srcEvent` pentru un obiect `event`, ați fi scris șase funcții de scriptare separate pentru fiecare nume de oraș, care să stabilească stilurile corespunzătoare de text pentru paragraf.

Listingul 29.4 arată un exemplu de chestionar interactiv, care arată minunat ca pagină Web. Utilizatorului i se pune o întrebare despre planetele sistemului solar și se evaluează răspunsul său când execută clic pe o imagine bitmap de planetă. Exemplul conține o singură întrebare, dar, cu HTML 4 dinamic, ați putea să transmiteți un examen întreg ca o singură pagină Web, să calculați și să prezentați rezultatele testului prin funcții script. Aruncați o privire către Listingul 29.4 și Figurile 29.5, 29.6, iar apoi continuați să citiți pentru detalii suplimentare.

Listingul 29.4. Un exemplu de executare de clic și înlocuire.

```
<HTML>
<HEAD>
<FONT FACE-"verdana,arial,helvetica" SIZE=2>
<TITLE>Planetary Trivia</TITLE>
</HEAD>
<BODY BGCOLOR="#FFFFFF">
<H2>Planetary Trivia</H2>
<HR>
<BR>
<H3><P ID=Q1 STYLE="color:Blue">
<B>Question 1: </B>
</H3><P>
Which of the following planets reflects the highest percentage
of light it receives from the sun?
<BR>
<TABLE>
<TR><TD><IMG SRC="mars.gif" ID=A11 onclick="Answer1();"></TD>
<TD><IMG SRC="venus.gif" ID=A12 onclick="Answer1();"></TD>
<TD><IMG SRC="saturn.gif" ID=A13 onclick="Answer1();"></TD>
<TD><IMG SRC="uranus.gif" ID=A14 onclick="Answer1();"></TD></TR>
</TABLE>
<P ID=A1></P>

<SCRIPT LANGUAGE=JavaScript>.
var srcElement
function Answer1() {
  srcElement = window.event.srcElement;
  if (Q1.style.color=="blue") {
     if (srcElement.id=="A12") {
        srcElement.src = "correct.gif";
        Q1.style.color = "Green";
     } else {
        srcElement.src = "wrong.gif";
        Q1.style.color = "Red";
    }
    A1.innerHTML="<B>Venus</B> has an albedo of 59% " +
       "which is much higher than the other answers";
  }
}
```



```
</SCRIPT>
<HR>
</BODY>
</HTML>
```

Figura 29.5 arată pagina Web cu informații planetare, imediat după ce se încarcă într-un browser Web. Utilizatorul trebuie să execute clic pe un răspuns.

FIGURA 29.5. *Înainte de declanșarea evenimentului* `onclick`.

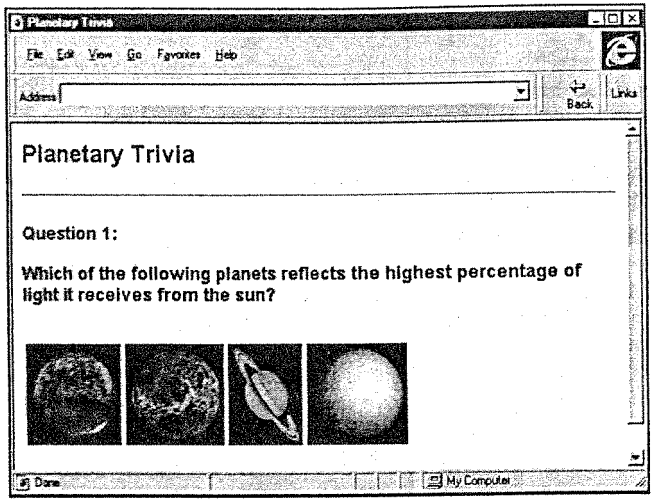
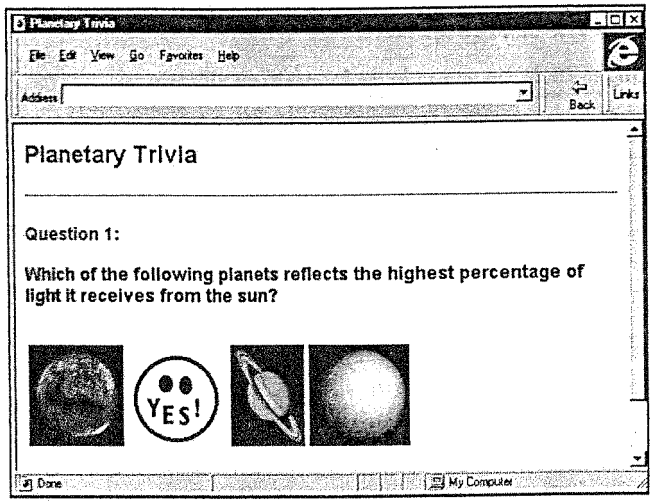


Figura 29.6 arată pagina Web cu informații planetare imediat după ce un utilizator a ghicit că Venus este răspunsul corect. Venus este într-adevăr răspunsul corect, deci utilizatorul este recompensat cu o imagine de față zâmbitoare. Culoarea titlului întrebării devine verde pentru a indica, și ea, un răspuns corect.

FIGURA 29.6. *După declanșarea evenimentului* `onclick`.



Numărul întrebării apare într-un paragraf inclus ca titlu, numit Q1, cu un atribut STYLE="color:Blue", care face albastru textul antetului întrebării. Imaginile lui Venus, Marte, Saturn și Uranus sunt furnizate în elemente cu eticheta <IMG> sub numele A11 la A14. Aceste imagini de planete includ, toate, atributul onclick="Answer1()", care configurează un eveniment să se declanșeze la un clic al utilizatorului pe element. Răspunsurile sunt evaluate în funcția script Answer1().

Ultima linie înainte de elementul script creează un paragraf de răspuns, unde apare text pentru a prezenta răspunsul corect. Paragraful este numit A1 în interiorul etichetei <P>, în atributul ID=A1.

Funcția script de tratare a evenimentului, Answer1(), este plasată în perechea de etichete <SCRIPT></SCRIPT>. Similar Listingului 29.3, scriptul folosește o variabilă numită source pentru a ține evidența planetei pe a cărei imagine utilizatorul a executat clic. Dacă paragraful antet pentru întrebare este de orice altă culoare decât albastru, scriptul nu face nimic, deoarece utilizatorul a făcut deja o încercare de a răspunde.

Dacă elementul sursă pentru eveniment este imaginea A12 (Venus), atunci scriptul înlocuiește imaginea lui Venus cu o imagine pentru răspuns corect. De asemenea, scriptul modifică în verde culoarea antetului întrebării. Dacă elementul sursă pentru eveniment este oricare altă imagine decât A12, scriptul înlocuiește imaginea selectată cu o imagine pentru răspuns greșit și modifică în roșu culoarea antetului întrebării. Fie în cazul unui răspuns corect, fie în cel al unui răspuns greșit, scriptul încheie prin a înlocui conținutul paragrafului pentru răspuns (A1) cu un text care prezintă răspunsul corect. Când scrieți cod pentru browserul Internet Explorer 4.0, scripturile înlocuiesc textul prin schimbarea proprietății innerHTML a oricărui element text.

Un exemplu de interfață cu utilizatorul

În mod obișnuit, pentru proiectarea unei aplicații efective contează foarte mult dacă interfața de utilizator este ușor de înțeles. HTML 4 vă oferă capacitatea de a vă proiecta propria interfață simplă de utilizator pe pagini Web. Exemplul din această secțiune adaugă o interfață simplă de utilizator formularului de comandă a biletelor de concert din Listingul 29.2. Listingul 29.5 furnizează conținutul fișierului HTML, iar Figura 29.7 arată fișierul încărcat într-un browser Web.

Listingul 29.5. Un exemplu de interfață cu utilizatorul.

```
<HTML>
<HEAD>
<FONT FACE="verdana,arial,helvetica" SIZE=2>
<TITLE>Form Example</TITLE>
</HEAD>
<BODY BGCOLOR="#FFFFFF">
<H1>Concert Ticket Order Form</H1>
Maximum 10 tickets per order
<HR>
<TABLE><TR><TD>
<BUTTON ID=1 onClick="FillQty();">1</BUTTON>
<BUTTON ID=2 onClick="FillQty();">2</BUTTON>
<BUTTON ID=3 onClick="FillQty();">3</BUTTON>
<BUTTON ID=4 onClick="FillQty();">4</BUTTON>
<BUTTON ID=5 onClick="FillQty();">5</BUTTON>
<BUTTON ID=6 onClick="FillQty();">6</BUTTON>
```



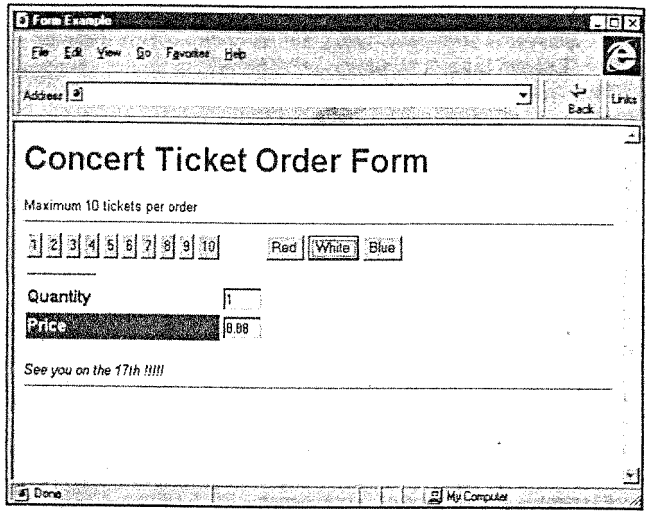
```
<BUTTON ID=7 onClick="FillQty();">7</BUTTON>
<BUTTON ID=8 onClick="FillQty();">8</BUTTON>
<BUTTON ID=9 onClick="FillQty();">9</BUTTON>
<BUTTON ID=10 onClick="FillQty();">10</BUTTON>
</TD><TD></TD><TD>
<BUTTON ID=Red onClick="Colorize();">Red</BUTTON>
<BUTTON ID=White onClick="Colorize();">White</BUTTON>
<BUTTON ID=Blue onClick="Colorize();">Blue</BUTTON>
</TD></TR><TR><TD>-------------</TD></TR>
<TR><TD ID=Qtylbl><B>Quantity</B></TD>
<TD><INPUT NAME="qty" VALUE="0" SIZE=4>
</TD></TR>
<TR><TD ID=Pricelbl><B>Price</B></TD>
<TD><INPUT NAME="price" VALUE="" SIZE=4>
</TD></TR>
</TABLE>
<P><I>See you on the 17th !!!!!</I>
<SCRIPT language=javascript>
function FillQty() {
  srcElement = window.event.srcElement;
  qty.value=srcElement.id;
  price.value=qty.value*8.88;
}
function Colorize() {
  srcElement = window.event.srcElement;
  Qtylbl.style.background=srcElement.id;
  Pricelbl.style.color=srcElement.id;
  if(srcElement.id=="White") {
       Pricelbl.style.background="Red";
  } else Pricelbl.style.background="White";
}
</SCRIPT>
<HR>
</BODY>
</HTML>
```

Figura 29.7 arată pagina Web de comenzi pentru concert după ce utilizatorul a executat clic pe două butoane: butonul cu eticheta 1 și butonul cu eticheta White. Asemănător Figurii 29.2, prețul total a fost calculat de scriptul de tratare a evenimentului și afișat în câmpul de text Price.

În Listingul 29.5, este creată o interfață de utilizator simplă din treisprezece elemente buton. Primele zece butoane au nume de la 1 la 10 și configurează un eveniment onclick prin atributul onclick="FillQty();". Următoarele trei butoane au numele Red, White și Blue (roșu, alb și albastru) și configurează un eveniment onclick prin atributul onclick="Colorize();".

Ambele funcții script, FillQty() și Colorize(), sunt plasate în interiorul aceleiași perechi de etichete <SCRIPT></SCRIPT>. Cele două funcții ar putea, la fel de ușor, să aibă propriile lor elemente script, dar dacă doriți să folosiți același limbaj de scriptare pentru două funcții script, de ce să nu le plasați în același element?

FIGURA 29.7. *Adăugarea unei interfețe de utilizator simple.*



OBSERVAȚIE

Amintiți-vă și că plasarea perechii de etichete `<SCRIPT></SCRIPT>` nu este limitată de elementele `HEAD` și `BODY`. Câtă vreme nu încălcați nici una dintre regulile de înglobare a etichetelor în etichete, plasarea scripturilor este foarte flexibilă.

Funcția script `FillQty()` umple atât câmpul de text `Quantity`, cât și câmpul de text `Price` când un utilizator selectează o cantitate executând clic pe un buton numerotat. Atributul `VALUE` al elementului de introducere `Qty` primește ca valoare numele `id` al butonului numerotat pe care s-a executat clic. Valoarea `VALUE` a elementului de intrare `Price` este calculată prin înmulțirea cantității cu 8,88 USD, prețul unui bilet.

Funcția script `Colorize()` schimbă culoarea etichetelor `Quantity` și `Price`. Mai întâi, scriptul înlocuiește culoarea de fond a elementului text `Qtylbl` cu numele `id` al butonului de culoare pe care s-a executat clic. Apoi, scriptul schimbă culoarea de text a elementului text `Pricelbl` în aceeași culoare ca și culoarea elementului text `Qtylbl`. Pentru a evita situația alb-pe-alb, scriptul schimbă culoarea de fond a elementului text `Pricelbl` dacă utilizatorul alege butonul White.

Etichetele <NOSCRIPT></NOSCRIPT>

Această secțiune tratează problemele generate de lipsa suportului pentru scriptare, pe care autorii ar trebui să le ia în considerare când proiectează documente HTML de valoare. Perechea de etichete `<NOSCRIPT></NOSCRIPT>` permite autorilor să prevadă un conținut alternativ pentru situația în care un script nu este executat. W3C menționează despre cuprinsul unui element `<NOSCRIPT>` că ar trebui redat numai de un agent de utilizator care recunoaște scripturi aflat într-unul din următoarele două cazuri:

- agentul de utilizator este configurat să nu evalueze scripturi,
- agentul de utilizator nu acceptă un limbaj de scriptare invocat de un element `SCRIPT` mai înainte, în document.

De asemenea, specificația HTML 4 cere agenților de utilizator care nu acceptă scripturi de partea clientului să redea conținutul perechii de etichete `<NOSCRIPT></NOSCRIPT>` ca și cum ar fi fost cuprinsul unui element `BODY`.

Listingul 29.4 oferă un exemplu bun de includere corespunzătoare a perechii de etichete `<NOSCRIPT></NOSCRIPT>`. După linia `<P ID-A1></P>`, pagina Web ține cont și de agenții de utilizator mai vechi prin adăugarea următoarei sintaxe:

```
<NOSCRIPT>
<P>
The correct answer is <B>Venus</B>, the second image from the left. Venus
has an albedo of 59% which is much higher than the other answers.
</NOSCRIPT>
```

Acest conținut ar fi transmis de un browser Web numai dacă una dintre cele două condiții specificate anterior ar fi adevărată. Desigur, fără un browser Web capabil de scripturi, utilizatorii nu ar putea da textul în mod interactiv, dar ar putea învăța din pagina Web, deoarece răspunsul ar fi prezentat împreună cu întrebarea.

Scripturi transformate în comentariu

Agenții de utilizator care nu recunosc perechea de etichete `<NOSCRIPT></NOSCRIPT>` vor reda aproximativ cuprinsul elementului script ca text. W3C dă prilejul unei discuții asupra comentariilor în documentul lor de lucru asupra HTML 4. Restul acestei secțiuni reproduce discuțiile W3C asupra transformării scripturilor în comentariu.

Unele motoare de scriptare, cum sunt cele pentru limbajele JavaScript, VBScript și TCL, permit includerea instrucțiunilor din script într-un comentariu SGML. Agenții de utilizator care nu recunosc elementele script vor ignora, în acest fel, comentariul, dar motoare de scriptare inteligente vor înțelege că scripturile din comentarii ar trebui executate.

O altă soluție a problemei este să păstrați scripturile în documente externe și să vă referiți la ele cu atributul `src`.

Motorul JavaScript permite apariția șirului `<--` la începutul unui element `SCRIPT` și ignoră caracterele ulterioare până la sfârșitul liniei. JavaScript interpretează `//` ca începutul unui comentariu care se extinde până la sfârșitul liniei curente. Acesta este necesar pentru a ascunde șirul `-->` de analizorul JavaScript.

```
<SCRIPT type="text/javascript">
<-- to hide script contents from old browsers
    function square(i) {
      document.write("The call passed ", i ," to the function.","<BR>")
      return i * i
  }
  document.write("The function returned ",square(5),".")
// end hiding contents from old browsers -->
</SCRIPT>
```

În VBScript, caracterul ' (apostrof) face ca restul liniei curente să fie tratată ca un comentariu. Prin urmare, poate fi utilizat să ascundă șirul `-->` de VBScript, ca în exemplul următor:

```
<SCRIPT type="text/vbscript">
  <--
    Sub foo()
      ...
```

```
    End Sub
  '-->
</SCRIPT>
```

În TCL, caracterul # transformă în comentariu restul liniei:

```
<SCRIPT type="text/tcl">
<-- to hide script contents from old browsers
  proc square(i) {
    document write "The call passed $i to the function.<BR>"
    return [expr $i * $i]
  }
  document write "The function returned [square5]."
# end hiding contents from old browsers -->
</SCRIPT>
```

OBSERVAȚIE

Unele browsere închid comentariile la primul caracter >, astfel încât, pentru a ascunde conținutul scriptului de astfel de browsere, puteți reformula operanzii operatorilor relaționali și de deplasare (de exemplu, folosiți y < x în loc de x > y) sau folosiți coduri escape dependente de limbajul de scriptare pentru >.

Rezumat

HTML-ul dinamic reprezintă o îmbunătățire a HTML 4 care adaugă modificări dinamice și interactivitate paginilor Web. Folosind sintaxa HTML-ului dinamic, autorii de pagini Web pot realiza aplicații care se ridică la nivelul titlurilor de CD-ROM și aplicațiilor tradiționale de afaceri, pentru a satisface cerințele actuale ale pieței. HTML-ul dinamic instanțiază elementele paginii Web ca obiecte prin intermediul atributului `ID` sau `NAME`. După ce un element este instanțiat, evenimentele de utilizator pot declanșa modificări ale elementului, precum schimbarea fontului, a clasei de stil, a dimensiunii, a culorii sau a locației. Evenimentele sunt configurate ca atribute ale elementului sursă al fiecărui eveniment. De exemplu, atributul `onload="Function1();"` configurează un eveniment `onload` în interiorul unei etichete `<BODY>`.

Capitolul 30
Asocierea datelor

de Bruce Campbell

În acest capitol

- Asocierea datelor și conștientizarea datelor 435
- De ce să folosim asocierea datelor? 436
- Distribuirea datelor în zilele noastre 437
- Rolul agentului de utilizator 437
- Formularul de date 438
- Adăugarea conștientizării datelor cu HTML dinamic 438
- Exemple de formulare Web conștiente de date 439
- Generarea dinamică a tabelelor 448
- Un exemplu de generare dinamică de tabel 450
- Un exemplu de sortare dinamică de tabel 453
- DATAFORMATAS și DATAPAGESIZE 456

Asocierea datelor și conștientizarea datelor

Un document este *conștient de date (data-aware)* dacă se asociază el însuși unei surse specifice de date, iar *asocierea datelor (data binding)* este procesul de conectare a unei surse de date la o pagină Web. Conștientizarea datelor este o caracteristică a aplicației de calculator care îi permite acesteia să acceseze date și să păstreze o referință a respectivei surse de date. Ca efect, aplicația continuă să fie la curent cu datele. Conștientizarea datelor prin asocierea datelor nu este parte a documentului de lucru al W3C (Consorțiul World Wide Web), dar este o componentă a tehnologiei HTML dinamic de la Microsoft. Asocierea datelor lucrează printr-un proces foarte asemănător procesului HTML de tratare a evenimentelor, drept care înțelegerea Capitolului 29, „HTML și scripturi – Combinația DHTML" este fundamentală pentru a înțelege materialul din acest capitol. Acest capitol tratează procesul de asociere a datelor.

Un joc video este conștient de date dacă are acces la clasament și stări anterioare ale jocului pe care le-ați salvat, în timp ce jucați. O astfel de asociere a datelor este o caracteristică importantă pentru un joc de aventură. Dacă jucați și realizați că ați făcut un pas greșit, puteți reîncărca o poziție anterioară din joc și să reluați din acel loc. Când un joc video este conștient de date, el poate accesa datele imediat și reface jocul într-o stare cerută.

Acum, imaginați-vă că vă distrați cu un joc bazat pe HTML dinamic dintr-o pagină Web și ați dori să fie descărcat o singură dată ca să ruleze local, pe mașina dumneavoastră. Dacă începeți să salvați stările jocului, nu trebuie să cereți serverului Web să realizeze aceste salvări pentru dumneavoastră. Un bun cetățean al Internetului știe că nu este necesar să creeze trafic Web fără a fi nevoie. În schimb, poate fi accesată o sursă de date asociate referitoare la salvarea stărilor dintr-o pagină Web și actualizată pe mașina client.

În cazul unui browser Web, conștientizarea datelor este caracteristica unui agent de utilizator care solicită toate datele posibile pentru o pagină Web o dată, se deconectează de la serverul Web și apoi interacționează cu datele local atâta vreme cât dorește utilizatorul. Asocierea datelor face o pagină Web conștientă de o sursă de date.

De ce să folosim asocierea datelor?

Pentru a înțelege de ce ar trebui să asociați surse de date la pagini Web HTML 4 conștiente de date, priviți un exemplu din realitate. Închipuiți-vă că faceți cumpărături pentru cadourile de Crăciun destinate familiei și prietenilor dumneavoastră. Dacă faceți un tur prin catalogul dumneavoastră favorit de comenzi prin poștă, sunteți uimit de cât de ușor găsiți zece articole care i-ar putea face fericiți pe cei iubiți. Ziua de naștere a sorei dumneavoastră vine pe 18 decembrie, iar pe 21 vă întâlniți cu rudele din partea soției.

Trebuie să luați o decizie. Pentru dumneavoastră nu este nici o diferență între a solicita trei transporturi diferite care să sosească la cele trei date diferite când aveți nevoie de cadouri (pe 18, 21 și 25) sau să le aveți pe toate transportate la aceeași dată. Totuși, trei curse diferite ale mașinii care face livrările înseamnă un consum suplimentar de resurse naturale, fără a menționa crearea de documente de lucru suplimentare (sau intrări în baza de date) pentru compania de comenzi prin poștă. I-ați putea face pe alții să își aștepte mai mult livrările, deoarece mașina oprește la domiciliul dumneavoastră înainte de a se duce la ei. Acțiunile dumneavoastră simple legate de o cumpărare prin ordin poștal au efect asupra multor altor lucruri și oameni.

Distribuirea datelor pe Web este asemănătoare distribuirii mărfurilor prin poștă. În zilele noastre, săriți de la o pagină la alta, iar serverul livrează datele din următoarea pagină de fiecare dată când executați clic pe o legătură. Toate exemplele din Capitolul 29 livrează datele în această manieră. Alți vizitatori Web așteaptă livrarea paginii lor (măcar că microsecunde), în timp ce serverul distribuie paginile altor persoane.

Cu un browser Web conștient de date, toate datele pot fi livrate la o singură vizită a serverului Web, iar browserul este la curent cu datele. Cu HTML dinamic, pagina Web poate să se modifice în mod dinamic pentru a prezenta datele bazându-se pe acțiuni interactive ale utilizatorului. De fapt, un cititor ar putea adăuga date noi sursei de date, dacă este necesar. Datele pot fi reîncărcate la sfârșitul folosirii paginii Web. De acum, ar trebui să vedeți beneficiile adăugării conștientizării datelor browserului Web și specificației HTML dinamic. Adăugați la toate acestea și beneficiul unei mai bune gestionări a întreținerii paginii Web. Autorii descoperă că întreținerea paginii Web este mult mai ușoară (și, de aceea, mai puțin scumpă) când datele variabile sunt menținute separat de porțiunile fixe de text și grafică ale paginii.

OBSERVAȚIE

Cel mai mare neajuns al asocierii datelor este că, adesea, sunt furnizate mai multe date decât ce este, în realitate, folosit de către destinatar. Datele suplimentare consumă lățimea de bandă a rețelei, care ar putea fi folosită pentru date care sunt, într-adevăr, consumate. Totuși, deoarece metodele de comprimare pentru surse de date se îmbunătățesc continuu, se pare că schimbarea merită, de obicei, osteneala din considerație pentru serverul Web.

Distribuirea datelor în zilele noastre

Astăzi, multe date sunt livrate consumatorilor la nivel de întreprindere prin intermediul rețelelor locale de calculatoare (LAN, local area networks) și majoritatea datelor la nivel de întreprindere sunt stocate în baze de date relaționale. Teoria bazelor de date relaționale a fost încorporată în mediile de calcul pentru capacitatea sa de a livra date eficient și neredundant. De obicei, datele sunt livrate spre un PC într-o manieră care poate fi eficient prezentată de un instrument front-end pentru baze de date relaționale, cum ar fi Access, Paradox sau FoxPro, pentru a numi numai câteva.

Consumatorii de volume mari de date pot solicita datele de la serverele de date prin interogare (query) cu ajutorul unui limbaj de interogare. *Interogare (query)* este doar un termen pretențios pentru o cerere de informații, într-un format specific, de la baza de date unde informațiile sunt stocate. Structured Query Language (SQL, limbaj structurat de interogare) este limbajul standard pentru cereri și manipulare de date relaționale. Aproape toate serverele de date înțeleg SQL. Un server de date acceptă interogarea, o rulează și trimite înapoi setul de rezultate ale interogării prin rețeaua de întreprindere. Doar de curând lumea afacerilor a luat în considerare folosirea Internetului ca parte a rețelei sale.

OBSERVAȚIE

Dacă vă gândiți la beneficiile unui mod standard, structurat de a cere informații din baza de date, începeți să observați toate beneficiile pe care standardul HTML le oferă în prezentarea Web. Cu HTML, puteți folosi una dintre multele aplicații de software de tip browser pentru a solicita o pagină Web de la oricare din multele aplicații server. Cu SQL, puteți folosi unul dintre multele instrumente de analiză a datelor pentru a solicita date de la oricare din multele aplicații server. Standardul (HTML sau SQL) este mijlocitorul unor îmbunătățiri tehnologice independente, din ambele părți.

Nu oricare consumator de date din lumea afacerilor cunoaște cum să solicite date în mod corect pentru a fi cât se poate de eficient în privința resurselor de calcul și de rețea partajate. Nici de la consumatorii publici de Web nu se poate aștepta să înțeleagă care sunt compromisurile necesare. De fiecare dată când solicit date relaționale dintr-o rețea mare, iau o decizie. Să solicit mai multe date decât am nevoie curent, încât să le am pe PC-ul meu când vine vremea să le folosesc? Sau să cer numai informațiile sumare de care am nevoie acum?

Deciziile de acces la date ne-au înconjurat dintotdeauna. Pentru comunitatea de afaceri, tehnicienii au lucrat la ambele margini ale spectrului pentru a livra ce este mai bun în ambele lumi. Instrumentele de analiză a datelor sunt gata să accepte toate datele în același timp și să furnizeze instrumentele de a le analiza pe bucăți, pe un PC. Serverele Web sunt optimizate să analizeze, calculeze și livreze simultan de la zeci la sute de cereri simultane de date. Întrucât tehnicienii au lucrat asupra acestor înlesniri fundamentale de zeci de ani, mulți cred că este just să ofere aceleași beneficii restului lumii de pe Web.

Cu cât mai mulți tehnicieni caută să aplice Webului beneficiile demonstrate în timp ale teoriei bazelor de date relaționale, cu atât mai mulți cititori se confruntă cu dorința de a avea ce este mai bun din ambele lumi în interiorul paginii Web. Din nou, ați ajuns la un punct unde ar trebui să luați în considerare paginile Web conștiente de date și browserele Web.

Rolul agentului de utilizator

Pentru ca o strategie de distribuire a unei pagini Web conștiente de date să funcționeze, un browser Web trebuie să fie capabil să înțeleagă formatul datelor pe care serverul Web le livrează. Browserul trebuie să așeze datele într-un loc în memorie unde să le poată regăsi

ușor. De asemenea, browserul Web trebuie să își amintească unde a pus datele când pagina Web dorește să afișeze mai multe dintre ele.

Browserul Web Internet Explorer poate trata primirea, plasarea, calculul și prezentarea solicitărilor paginilor Web conștiente de date. Pentru Microsoft, aș spune că adăugarea funcționalității necesare de date conștiente nu a fost foarte dificilă, deoarece crease de câțiva ani instrumentul de analiză a datelor Access și dezvoltase procesul de asociere a datelor pentru mediul de aplicații Visual Basic. Cu toate acestea, fără HTML dinamic, adăugarea conștientizării datelor paginilor Web ar fi stângace, deoarece nu ar exista nici o cale simplă de a comunica dorința unui cititor de a avea mai multe date către sursa de date care le-a livrat.

Modelul de evenimente HTML dinamic oferă mai mult decât suficientă interactivitate pe o pagină Web, astfel încât să asigure strategii clare, ușor de înțeles de prezentare și modificare a datelor. Puteți executa clic pe un buton pentru a vedea următorul rând de date din sursa datelor. Puteți introduce o dată calendaristică într-o casetă de text pentru a vedea informațiile pentru o zi anume. Puteți trece dispozitivul dumneavoastră de indicare peste un fișier de imagine anume pentru a solicita datele respectivei imagini. Puteți executa clic pe un tabel cu date pentru a sorta tabelul după rândul respectiv. Cu alte cuvinte, puteți efectua multe dintre aceleași funcții de analiză a datelor ca și în instrumentele populare de manipulare a datelor.

Formularul de date

Creați un formular de date pentru a prezenta date într-o pagină Web. Un *formular de date (data form)* reprezintă un obiect de prezentare care conține etichete ce definesc semnificația fiecărei date și câmpuri în care se afișează datele unei anumite părți a sursei de date. Uzual, o parte a sursei de date care se referă la un obiect specific tip date este numită *înregistrare (record)*. Dumneavoastră folosiți un formular de date pentru a prezenta câte o înregistrare la un moment dat. Afișarea doar a unui subset de înregistrări o dată este numită *filtrare (filtering)*. HTML-ul dinamic dă cititorilor posibilitatea de a indica dacă ar vrea să vadă o înregistrare diferită.

Formularele de date pot să arate foarte încântător dacă adăugați grafică și culoare folosind etichetele HTML adecvate. Astfel, fiecare înregistrare poate fi prezentată într-un mod care seamănă cu paginile Web tradiționale.

Adăugarea conștientizării datelor cu HTML dinamic

Până acum, acest capitol a constat, în principal, din definiții și viziuni. Analiștii de date născocesc mereu termeni noi și se gândesc la viitor; și le place să facă aceasta. Această secțiune tratează detalii ale modului de a adăuga o sursă de date unei pagini Web HTML destinată vizualizării cu browserul Web Internet Explorer 4.0.

Pentru a adăuga conștientizarea datelor unei pagini Web, folosiți o pereche de etichete `<OBJECT></OBJECT>` cu toate atributele corespunzătoare. Priviți un exemplu:

```
<OBJECT ID="townnews"
     CLASSID="clsid:333C7BC4-460F-11D0-BC04-0080C7055A83"
     BORDER="0" WIDTH="0" HEIGHT="0">
     <PARAM NAME="DataURL" VALUE="towndata.txt">
     <PARAM NAME="UseHeader" VALUE="True">
</OBJECT>
```


Eticheta de deschidere `<OBJECT>` conține un atribut `ID` care specifică numele sursei de date. În acest caz, sursa de date conține știri pentru mai multe orașe. Atributul `CLASSID` face trimitere la controlul ActiveX care oferă conștientizarea datelor unei pagini Web în Internet Explorer 4.0. Controlul ActiveX pentru conștientizarea datelor este livrat împreună cu Internet Explorer 4.0.

Observați că obiectul sursă de date nu ocupă nici un spațiu pe pagina Web; în schimb, componente ale sursei de date sunt accesate și prezente în alte etichete din pagina Web. Etichetele imbricate `<PARAM>` și atributele lor sunt cerute întrucât browserul Web folosește parametrii pentru a configura corect sursa de date. Parametrul `NAME=DataURL` furnizează o valoare (`VALUE`) care identifică numele de fișier al sursei de date. Parametrul `UseHeader` specifică dacă se folosește primul rând din sursa de date ca antet pentru restul înregistrărilor din sursa de date. O valoare (`VALUE`) adevărată (`True`) comunică browserului Web să se refere la fiecare coloană din sursa de date prin numele din fruntea coloanei.

Pentru a crea formularul de date în pagina dumneavoastră Web, asociați o etichetă și un câmp de date cu fiecare coloană din sursa dumneavoastră de date. Următorul exemplu creează o etichetă definită cu perechea de etichete `<LABEL></LABEL>`:

```
<LABEL FOR=hightemp>Today's High Temperature: </LABEL>
```

Eticheta de deschidere `<LABEL>` conține un atribut `FOR` care precizează câmpul pentru care eticheta este destinată. Puneți textul pentru etichetă între etichetele `<LABEL></LABEL>`.

Creați un câmp așa cum se vede mai jos, unde o etichetă `<INPUT>` conține atributele corespunzătoare care creează câmpul de formular:

```
<INPUT ID=hightemp TYPE=text DATASRC=#townnews DATAFLD="High">
```

Atributul `ID` denumește câmpul. Atributul `TYPE` definește tipul datelor pe care câmpul le păstrează. Atributul `DATASRC` asociază cîmpul cu coloana de date corectă în sursa de date.

Exemple de formulare Web conștiente de date

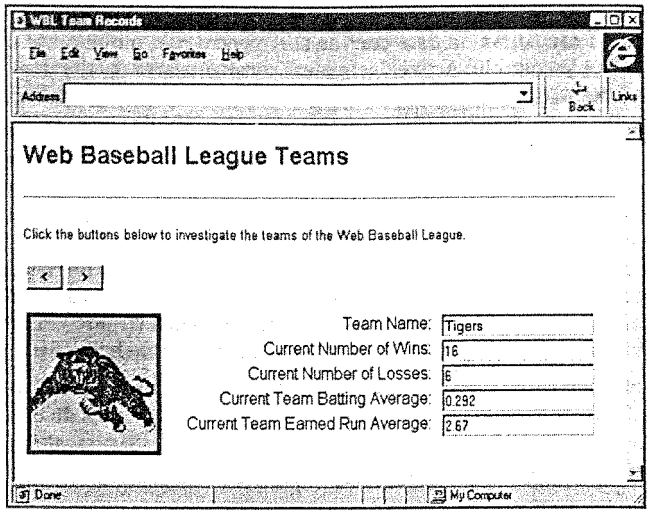
Acest capitol vă oferă două exemple de formulare Web conștiente de date pe care le puteți îngloba într-o pagină Web. Primul arată o pagină sportivă care este organizată în jurul conceptului de vestiar cu dulapuri. Cititorul poate indica și executa clic pe pagina Web pentru a vizita vestiarul fiecărei echipe și a vedea statisticile referitoare la respectiva echipă. Al doilea exemplu arată o pagină Web geografică interactivă.

Un prim exemplu de asociere a datelor

În exemplul din Listingul 30.1, camerele vestiarului sunt paginile pentru echipe din Web Baseball League, un campionat de baseball fictiv, jucat pe Web. Fiecare pagină vestiar arată numărul de jocuri câștigate și pierdute, media loviturilor la bătaie și raportul victoriilor (coeficientul ERA) după ultimul joc susținut de fiecare echipă din campionat. Inițial, când pagina se încarcă, este prezentat vestiarul echipei Tigers, așa cum se vede în Figura 30.1. Cititorii pot folosi cele două butoane, marcate > și <, pentru a trece înainte sau înapoi prin diferitele vestiare.

Aproape toți furnizorii importanți de informații sportive pe Web folosesc un concept de vestiar, astfel încât cititorii să poată naviga direct către echipa lor favorită. Dacă, apoi, un cititor decide să meargă în vestiarul altei echipe, se face o a doua cerere către serverul Web pentru a furniza informațiile pentru respectiva echipă. În cazul exemplului din Listingul 30.1, datele sunt livrate însă toate o dată, într-un singur fișier. Când cititorii răsfoiesc printre diferitele echipe, ei nu trebuie să depindă de nici o livrare de date adițională din Web.

FIGURA 30.1.
Vestiarul echipei Tigers.



Priviți Listingul 30.1, iar apoi continuați să citiți pentru explicații detaliate. Figura 30.2 arată vestiarul echipei Frogs, așa cu este afișat în Internet Explorer 4.0.

Listingul 30.1. Un exemplu de prezentare dinamică a înregistrărilor.

```
<HTML>
<HEAD>
<FONT FACE="verdana,arial,helvetica" SIZE=2>
<TITLE>WBL Team Records</title>
</HEAD>
<BODY BGCOLOR="#FFFFFF">
<H2>Web Baseball League Teams</H2>
<HR>
<P>Click the buttons below to investigate the teams of the Web Baseball
League.</P>
<P>
<OBJECT ID="teamlist"
        CLASSID="clsid:333C7BC4-460F-11D0-BC04-0080C7055A83"
        BORDER="0" WIDTH="0" HEIGHT="0">
  <PARAM NAME="DataURL" VALUE="wblteam.txt">
  <PARAM NAME="UseHeader" VALUE="True">
</OBJECT>

<TABLE>
  <TR>
    <TD ALIGN=RIGHT><INPUT TYPE=BUTTON ID=backward VALUE=" < "></TD>
    <TD ALIGN=LEFT><INPUT TYPE=BUTTON ID=forward VALUE=" > "></TD>
  </TR>
</TABLE>
<P>
<IMG ID=Picture SRC="Tigers.jpg" BORDER=4 HEIGHT=128 WIDTH=128 ALIGN=LEFT>

<TABLE ALIGN=CENTER CELLSPACING=0 CELLPADDING=0>
<TR>
```



```
<TD ALIGN=RIGHT VALIGN=TOP><LABEL FOR=team>Team Name: </LABEL></TD>
<TD ALIGN=LEFT VALIGN=TOP WIDTH="10"></TD>
<TD ALIGN=LEFT VALIGN=TOP><INPUT ID=team TYPE=text
DATASRC=#teamlist DATAFLD="Team"></TD>
</TR>

<TR>
<TD ALIGN=RIGHT VALIGN=TOP><LABEL FOR=wins>
Current Number of Wins: </LABEL></TD>
<TD ALIGN=LEFT VALIGN=TOP WIDTH="10"></TD>
<TD ALIGN=LEFT VALIGN=TOP><INPUT ID=wins TYPE=text
DATASRC=#teamlist DATAFLD="Wins"></TD>
</TR>

<TR>
<TD ALIGN=RIGHT VALIGN=TOP><LABEL FOR=losses>
Current Number of Losses: </LABEL></TD>
<TD ALIGN=LEFT VALIGN=TOP WIDTH="10"></TD>
<TD ALIGN=LEFT VALIGN=TOP><INPUT ID=losses TYPE=text
DATASRC=#teamlist DATAFLD="Losses"></TD>
</TR>

<TR>
<TD ALIGN=RIGHT VALIGN=TOP><LABEL FOR=BA>
Current Team Batting Average: </LABEL></TD>
<TD ALIGN=LEFT VALIGN=TOP WIDTH="10"></TD>
<TD ALIGN=LEFT VALIGN=TOP><INPUT ID=BA TYPE=text
DATASRC=#teamlist DATAFLD="BA"></TD>
</TR>

<TR>
<TD ALIGN=RIGHT VALIGN=TOP><LABEL FOR=ERA>
Current Team Earned Run Average: </LABEL></TD>
<TD ALIGN=LEFT VALIGN=TOP WIDTH="10"></TD>
<TD ALIGN=LEFT VALIGN=TOP><INPUT ID=ERA TYPE=text
DATASRC=#teamlist DATAFLD="ERA"></TD>
</TR>
</TABLE>

<SCRIPT LANGUAGE=JavaScript>
function documentClick() {
   Picture.src = team.value + ".jpg";
}

document.onclick = documentClick;

</SCRIPT>

<SCRIPT LANGUAGE=JavaScript>
function backwardClick() {
  if (teamlist.recordset.AbsolutePosition > 1) {
    teamlist.recordset.MovePrevious();
  } else {
    alert("Already at first team");
  }
}

backward.onclick = backwardClick;
```

(continuare)

Listingul 30.1. – Continuare

```
function forwardClick() {
  if (teamlist.recordset.AbsolutePosition != teamlist.recordset.RecordCount) {
    teamlist.recordset.MoveNext();
  } else {
    alert("Already at last team");
  }
}
forward.onclick = forwardClick;
</SCRIPT>
</FONT>
</BODY>
</HTML>
```

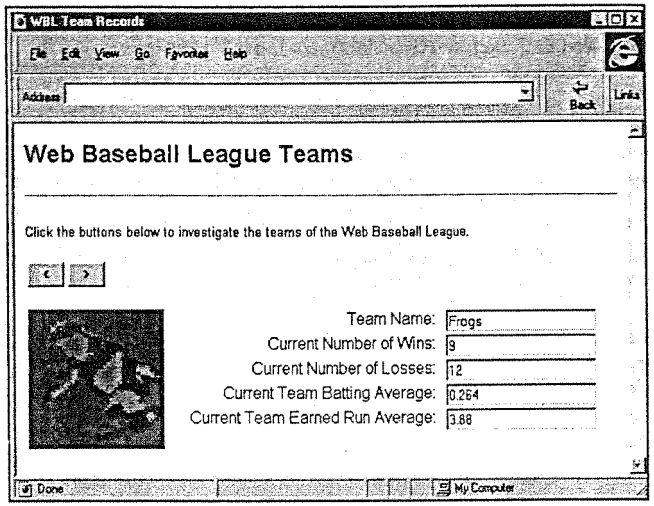


FIGURA 30.2. *O prezentare dinamică a vestiarului echipei Frogs.*

Obiectul de control al datelor folosit în acest exemplu este un control ActiveX pe care îl puteți descărca din Internet pentru a extinde posibilitățile browserului dumneavoastră Web. De fapt, Internet Explorer 4.0 include controlul pentru conștientizarea datelor în pachetul său standard. Același control este folosit pentru toate exemplele conștiente de date din acest capitol. Obiectul de control al datelor este inclus în pagina HTML cu perechea de etichete <OBJECT></OBJECT>. Liniile care identifică controlul ActiveX în Listingul 30.1 sunt:

```
<OBJECT ID="teamlist"
        CLASSID="clsid:333C7BC4-460F-11D0-BC04-0080C7055A83"
        BORDER="0" WIDTH="0" HEIGHT="0">
  <PARAM NAME="DataURL" VALUE="wblteam.txt">
  <PARAM NAME="UseHeader" VALUE="True">
</OBJECT>
```

Valoarea atributului ID instanțiază teamlist ca obiect sursă a datelor. Valoarea atributului CLASS_ID din eticheta <OBJECT> este un șir sofisticat cu securitate înglobată folosit la selectarea controlului ActiveX corespunzător pentru gestionarea sursei de date. Controlul

cere doi parametri, identificați în etichetele <PARAM> cu atributele NAME și VALUE. Parametrul DataURL furnizează browserului Web adresa URL a sursei datelor. Parametrul UseHeader precizează dacă rândul de antet al setului de date trebuie să fie utilizat ca informație pentru numele coloanei. Un tabel cu două butoane este adăugat paginii Web spre a oferi utilizatorului controale de mutare de la echipă la echipă. Ambele sunt instanțiate ca obiecte cu atributul ID. Un buton este instanțiat ca backward, celălalt ca forward.

Imaginea bitmap pentru emblema echipei Tigers este așezată pe pagină într-o etichetă <IMG> instanțiată ca un obiect cu atributul ID=Picture. Imaginea emblemei echipei se modifică prin intermediul funcției document_onclick(), declarată ulterior printr-un script. Informațiile despre echipa Tigers constituie primul rând al sursei de date wblteam.txt. Listingul 30.2 arată conținutul fișierului wblteam.txt.

Listing 30.2. Conținutul fișierului wblteam.txt.

```
Team,Wins:INT,Losses:INT,BA:FLOAT,ERA:FLOAT
Tigers,16,6,.292,2.67
Bears,14,8,.244,2.87
Robins,13,8,.267,2.67
Lions,11,10,.282,4.07
Rhinos,10,10,.255,3.55
Frogs,9,12,.264,3.88
Sharks,8,12,.251,4.15
Spiders,4,14,.212,3.86
```

Prima linie din Listingul 30.2 conține informațiile de antet pentru sursa de date. Team este numele primului câmp, iar Wins este numele celui de al doilea. Câmpurile Wins și Losses sunt de tip INT (întreg), iar câmpurile BA și ERA sunt de tip FLOAT (virgulă mobilă). Tipurile de date sunt folosite de browserul Web când prezintă datele cititorului. Următoarele linii din fișierul sursă de date reprezintă detaliile referitoare la fiecare echipă, păstrate ca înregistrări separate.

OBSERVAȚIE

Poate ați observat că fișierul wblteam.txt este într-un format simplu, de valori separate prin virgule (CSV, comma-separated values). Cu toată experiența pe care Microsoft a căpătat-o de-a lungul anilor în formate de fișier, puteți fi sigur că veți avea o mulțime de opțiuni pentru formatul de fișier al datelor furnizate unui surfer Web. Eu folosesc CSV deoarece este foarte simplu și ușor de folosit pentru depanare și pentru că aproape toate aplicațiile personale de date permit exportarea unui tabel bază de date în acest format.

După ce ați adăugat paginii imaginea cu emblema echipei, creați formularul de date. Formularul de date al echipei are cinci etichete și cinci câmpuri. Creați fiecare emblemă cu o pereche de etichete <LABEL></LABEL>, după cum urmează:

```
<LABEL FOR=team>Team Name: </LABEL>
```

Fiecare câmp este creat cu o etichetă <INPUT>, astfel:

```
<INPUT ID=team TYPE=text DATASRC=#teamlist DATAFLD="Team">
```

Etichetele pentru fiecare emblemă și câmp al formularului de date sunt similare. Pentru câmpul echipei, atributul TYPE identifică tipul de prezentare pentru câmp. Atributul DATASRC identifică sursa datelor (care are aceeași valoare cu numele obiectului sursă de date care indică fișierul wblteam.txt). Atributul DATAFLD precizează câmpul din sursa de date de la care caseta de introducere așteaptă să primească date. Numele câmpului se obține din primul rând al fișierului wblteam.txt.

OBSERVAȚIE

Pentru exemplul din Listingul 30.1, am optat să prezint formularul de date într-un tabel pentru un aspect mai organizat. Etichetele de tabel sunt cele din specificația HTML 3.2. Listingul 30.3 vă oferă un exemplu de metodă mai puțin organizată de a prezenta datele.

Prima pereche de etichete `<SCRIPT></SCRIPT>` include un script cu funcția `documentClick()`, disponibilă oricărei pagini Web încărcate într-un agent de utilizator care suportă HTML dinamic. În acest caz, funcția `documentClick()` este folosită pentru a păstra emblema echipei din vestiar în concordanță cu rândul de date curent care este arătat pe pagina Web. Această linie tratează fiecare clic de mouse de pe pagină pentru fiecare echipă:

```
Picture.src = team.value + ".jpg";
```

Ea schimbă atributul `SRC` pentru obiectul `Picture` bazându-se pe câmpul `Team` din șirul curent de date. Câmpul `Team` este instanțiat cu atributul `ID=team` al etichetei sale `<INPUT>`.

În interiorul celeilalte perechi de etichete `<SCRIPT></SCRIPT>`, două funcții tratează clicurile de buton ale utilizatorului. Funcția `backwardClick()` mută utilizatorul în sus în setul de date, cu un rând o dată, până ce este întâlnit primul rând; funcția `forwardClick()` mută utilizatorul în jos în setul de date, cu un rând o dată, până ce este întâlnit ultimul rând. Numele de funcții *trebuie* să înceapă cu același șir ca obiectele instanțiate existente. Dacă nu se folosesc exact aceleași șiruri pentru `backward` și `forward`, funcția ar putea să nu lucreze cum se intenționa.

Un al doilea exemplu

Al doilea exemplu arată un sit Web geografic, în care datele sunt prezentate mai mult ca o înșiruire de text obișnuit, decât într-un tabel curat și ordonat. În loc de deplasarea secvențială printre date cu o înregistrare o dată, cititorul poate executa clic pe o imagine pentru a vedea informațiile despre acel subiect particular. Formularul de date prezintă înregistrarea particulară, legată de regiunea selectată.

Acest exemplu folosește cele cinci state din Statele Unite aflate în nord-vestul Pacificului. Figura 30.3 arată prezentarea pentru statul Washington. Utilizatorul poate vedea informațiile pentru Washington executând clic pe imaginea statului Washington, care este parte a hărții.

OBSERVAȚIE

Pentru a obține cifrele referitoare la populația și suprafața Statelor Unite am vizitat situl Web de recensământ al guvernului Statelor Unite. Există atât de multe informații disponibile online la `http://www.census.gov/`, încât este copleșit chiar și un pătimaș al statisticilor, ca mine. Am folosit facilitățile pentru hărți de la `http://www.census.gov/datamap/www/` pentru a obține statisticile necesare în acest exemplu.

Dacă vizitați situl Web despre recensământ, observați cum părțile sunt organizate ca formulare de date din care solicitați anumite informații și așteptați să vă fie servite. Din septembrie 1997, situl Web de recensământ folosește încă scriptarea CGI ca să comunice cu un cititor de CD-ROM de pe serverul său. Prin urmare, situl Web de recensământ este un exemplu important al vechiului mod de interacțiune cu o sursă de date.

Sintaxa HTML-ului dinamic necesară pentru a crea exemplul de asociere a datelor din Listingul 30.3 este asemănătoare celei din Listingul 30.1. Aruncați o privire Listingului 30.3 și Figurii 30.3 și, apoi, continuați să citiți pentru mai multe detalii.

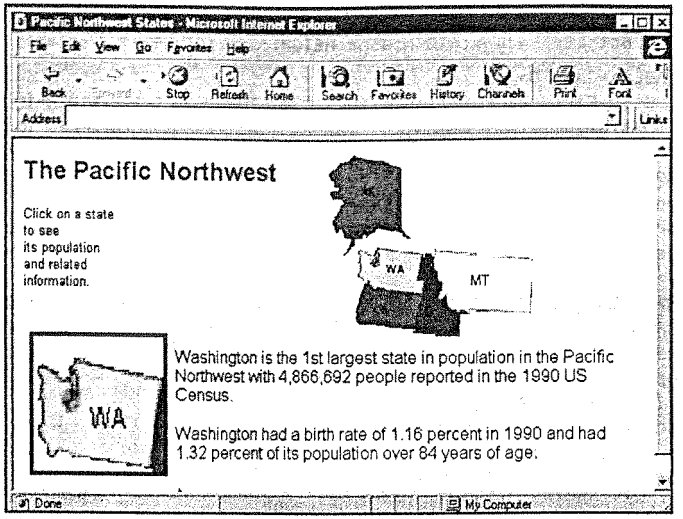


FIGURA 30.3. *Prezentarea statului Washington.*

Listingul 30.3. O prezentare de înregistrări dinamică, de tipul indică și execută clic.

```
<HTML>
<HEAD>
<FONT FACE="verdana,arial,helvetica" SIZE=2>
<TITLE>Pacific Northwest States</title>
</HEAD>
<BODY BGCOLOR="#FFFFFF">
<H2>The Pacific Northwest</H2>

<P ID=ne>Click on a state <BR>
to see <BR>
its population <BR>
and related <BR>
information.</P>
<P>
<OBJECT ID="statelist"
        CLASSID="clsid:333C7BC4-460F-11D0-BC04-0080C7055A83"
        BORDER="0" WIDTH="0" HEIGHT="0">
  <PARAM NAME="DataURL" VALUE="pacnw.txt">
  <PARAM NAME="UseHeader" VALUE="True">
</OBJECT>
<P>
  <IMG ID="alaska" STYLE="container:positioned;position:absolute;
TOP:10pt;LEFT:300px;WIDTH:96px;HEIGHT:96px;ZINDEX:1;" src="alaska.jpg">
  <IMG ID="idaho" STYLE="container:positioned;position:absolute;
TOP:80pt;LEFT:390px;WIDTH:48px;HEIGHT:84px;ZINDEX:5;" src="idaho.jpg">
  <IMG ID="washingt" STYLE="container:positioned;position:absolute;
TOP:70pt;LEFT:338px;WIDTH:64px;HEIGHT:48px;ZINDEX:4;" src="washingt.jpg">
  <IMG ID="oregon" STYLE="container:positioned;position:absolute;
TOP:100pt;LEFT:338px;WIDTH:64px;HEIGHT:48px;ZINDEX:2;" src="oregon.jpg">
  <IMG ID="montana" STYLE="container:positioned;position:absolute;
```

(continuare)

Listingul 30.3. – continuare

```
TOP:76pt;LEFT:410px;WIDTH:96px;HEIGHT:64px;ZINDEX:3;" src="montana.jpg">

<TABLE><TR><TD>
<IMG ID=Picture SRC="alaska.jpg" BORDER=4 HEIGHT=128 WIDTH=128 ALIGN=RIGHT>
</TD><TD>
<SPAN ID=state DATASRC=#statelist DATAFLD="State"></SPAN>is the
<SPAN ID=pr DATASRC=#statelist DATAFLD="Pop_Rank"></SPAN> largest state in
population in the Pacific Northwest with
<SPAN ID=pop DATASRC=#statelist DATAFLD="Population"></SPAN> people reported in
the 1990 US Census.
<P>
<SPAN ID=state2 DATASRC=#statelist DATAFLD="State"></SPAN> had a birth rate of
<SPAN ID=ar DATASRC=#statelist DATAFLD="Birth_Rate"></SPAN> percent in 1990 and
had <SPAN ID=area DATASRC=#statelist DATAFLD="Over_84"></SPAN> percent of its
population over 84 years of age.
</TD></TR></TABLE>
<SCRIPT language=JavaScript>
var srcElement
function documentClick() {
  srcElement = window.event.srcElement;
  if (srcElement.id=="alaska") {
    statelist.recordset.AbsolutePosition = 1;
  } else {
    if (srcElement.id=="washingt") {
      statelist.recordset.AbsolutePosition = 2;
    } else {
      if (srcElement.id=="oregon") {
        statelist.recordset.AbsolutePosition = 3;
      } else {
        if (srcElement.id=="idaho") {
          statelist.recordset.AbsolutePosition = 4;
        } else {
          if (srcElement.id=="montana") {
            statelist.recordset.AbsolutePosition = 5;
          }
        }
      }
    }
  }
  Picture.src = srcElement.id + ".jpg";
}
document.onclick = documentClick;

</SCRIPT>
</FONT>
</BODY>
</HTML>
```

Perechea de etichete <OBJECT></OBJECT> care configurează sursa de date este identică perechii din Listingul 30.1, exceptând faptul că numele și fișierul sursei de date sunt schimbate în statelist și respectiv pacnw.txt.

Imaginile celor cinci state sunt incluse prin etichetele <IMG>, fiecare cu un atribut SRC care identifică adresa URL a fiecărui fișier de imagine, un atribut ID care dă fiecărei imagini de obiect un nume și un atribut STYLE care plasează și dimensionează fiecare imagine pe pagina Web.

Asemănător Listingului 30.1, se adaugă paginii un obiect imagine mai mare pentru a furniza imaginea statului pe care cititorul îl selectează cu un clic de mouse. Inițial, imaginea este stabilită la statul Alaska. Dacă cititorul execută clic pe un stat diferit, imaginea mai mare este înlocuită cu cea a statului respectiv.

Fiecare câmp de înregistrare de date este configurat ca un câmp de formular separat pe pagina Web. Pentru formularele de date cu statele din nord-vestul Pacificului, folosesc perechea de etichete <SPAN></SPAN>, în locul perechii <INPUT></INPUT>, pe care am folosit-o în Listingul 30.1. Diferența este că în Listingul 30.3 câmpurile formularului de date sunt plasate sub forma unui text obișnuit cu curgere liberă, nu ca tabel. Amintiți-vă că diferența între etichetele de diviziune și cele de interval este că etichetele de interval nu includ un salt la linie nouă (line feed). Deoarece creați o prezentare cu curgere liberă, nu este nevoie să creați tabele.

Funcția script care leagă totul împreună este realizată cu caracteristica scrElement din Capitolul 29. Determin elementul sursă al unui clic de mouse de la cititor cu linia următoare din funcția documentClick():

```
srcElement = window.event.srcElement;
```

Funcția documentClick() este totdeauna disponibilă ca parte din HTML-ul dinamic.

Apoi, se determină atributul ID al imaginii selectate de cititor. Dependent de imaginea pe care cititorul execută clic, înregistrarea din sursa de date este stabilită la rândul respectiv cu o linie din funcție ca următoarea:

```
statelist.recordset.AbsolutePosition = 2.
```

Aici, indicatorul de rând este stabilit la rândul 2 (poziția sa absolută), care conține informația pentru statul Washington. Sunt create cinci astfel de linii în funcția script, deoarece există cinci state.

La sfârșit, sursa imaginii mai mari este stabilită la fișierul de imagine bitmap al statului pe care cititorul îl selectează. Pentru a schimba imaginea, se folosește următoarea linie:

```
Picture.src = state.value + ".jpg";
```

Datele specifice nord-vestului Pacificului vin de la fișierul sursă de date numit pacnw.txt. Listingul 30.4 arată conținutul fișierului. Observați că formatul fișierului este CSV (comma-separated values, valori separate prin virgule). De asemenea, observați că șirurile de date includ un caracter spațiu la sfârșit pentru a asigura o spațiere adecvată pe pagina Web.

Listingul 30.4. Conținutul fișierului pacnw.txt.

```
State,Pop_Rank,Population,Birth_Rate,Over_84
Alaska ,5th ,"550,043 ",".23 ","1.78 "
Washington ,1st ,"4,866,692 ","1.16 ","1.32 "
Oregon ,2nd ,"2,842,321 ","1.37 ","1.24 "
Idaho ,3rd ,"1,006,749 ","1.13 ","1.41 "
Montana ,4th ,"799,065 ","1.34 ","1.28 "
```

OBSERVAȚIE

Toate exemplele din acest capitol au utilizat formulare care permit numai prezentarea datelor. În Internet Explorer 4.0, Microsoft a implementat o arhitectură care vă permite să creați pagini Web cu formulare de date pe care cititorii dumneavoastră le pot folosi, la fel de bine, pentru a actualiza datele. Arhitectura HTML dinamic a Microsoft-ului include evenimente de validare a câmpului de formular, pe care le puteți folosi pentru a verifica tot conținutul unui formular bazat pe Web, înainte sau după ce actualizați o bază de date. Evenimentele de validare se bazează pe lista evenimentelor intrinseci din Capitolul 28.

Generarea dinamică a tabelelor

Expansiunea dinamică a tabelelor (Dynamic table expansion) este caracteristica unui instrument de prezentare a datelor care permite surselor de date de lungime variabilă să fie automat afișate într-un tabel de orice lungime. Când utilizați expansiunea dinamică a tabelelor într-o prezentare cu browserul Web, trebuie să identificați numai primul rând al tabelului cu anteturile corespunzătoare și apoi să precizați care coloană din sursa de date cu care câmp de antet merge. Browserul Web face restul, creând, în mod dinamic, etichetele de tabel necesare spre a arăta conținutul complet al sursei de date.

Caracteristica de expansiune dinamică a tabelelor lucrează bine, în special cu surse de date de lungime variabilă care se schimbă în timp. Nu trebuie să modificați pagina Web, deoarece browserul Web face toată munca de creare a etichetelor corespunzătoare conform numărului de înregistrări din sursa de date. Numai sursa datelor se schimbă pentru a prezenta cele mai recente date.

De exemplu, imaginați-vă că oferiți o „Pagină a zăpezilor proaspete" pe un sit Web de raport zilnic pentru schi. În fiecare zi, listați într-un tabel pe o pagină Web toți munții pe care a nins în ultimele douăzeci și patru de ore. Pentru fiecare munte, există o coloană cu numele muntelui, o coloană cu înălțimea de zăpadă proaspăt căzută și o coloană cu înălțimea totală a stratului de zăpadă. Pentru a prezenta informațiile în fiecare zi, ați putea beneficia de avantajele expansiunii dinamice a tabelelor pe o pagină Web conștientă de date. Fișierul HTML pe care îl creați poate să rămână același zi după zi. Numai sursa datelor care include un număr variabil de înregistrări (dependent de numărul munților pe care a căzut zăpadă în ultimele douăzeci și patru de ore) trebuie modificată zilnic. Browserul Web știe cum să prezinte informații variabile într-un tabel, bazându-se pe primul rând pe care îl identificați cu etichetele HTML corespunzătoare.

Folosiți expansiunea dinamică a tabelelor pentru simplitatea și flexibilitatea ei în prezentarea datelor în schimbare. Cu această caracteristică, puteți folosi surse de date multiple pe aceeași pagină Web prin repetarea codului pentru fiecare sursă de date. De exemplu, poate că doriți să arătați pe o pagină Web clasamentele zilnice din baseball pentru toate echipele de categorie Major League, AAA sau AA din Statele Unite și Canada. În fiecare noapte, fiecare secretar de ligă v-ar putea trimite o sursă de date cu pozițiile echipelor din respectiva ligă.

Zilnic, veți înlocui pe serverul Web fiecare fișier sursă pentru date cu un fișier cu același nume, iar cititorul care descarcă pagina de clasamente va obține ultimele poziții pentru fiecare echipă.

Expansiune dinamică înseamnă că fiecare clasament de ligă este deja expandat și vizibil pe pagină când browserul Web termină încărcarea paginii Web. Browserul Web Internet explorer 4.0 aduce și posibilitatea de sortare dinamică a datelor într-un tabel generat dinamic.

Browserul Web citește definiția primului rând din tabel din etichetele HTML <TABLE>, <TR> și <TD> care definesc rândul. Apoi, folosind atributele asociate cu un al doilea rând de etichete de tabel, browserul Web asociază coloanele potrivite din sursa de date cu diferitele celule din tabel. După aceea, browserul citește de-a lungul datelor și expandează tabelul pentru a face loc fiecărui rând de date din sursa de date.

Browserul Web trebuie să conțină logica de translatare necesară pentru a interpreta tipurile diferite de fișiere sursă de date. De asemenea trebuie să conțină și o logică oarecare de sortare dinamică a sursei datelor înainte de afișarea în tabel.

Pentru a adăuga expansiune dinamică de tabele paginilor dumneavoastră Web, inserați în fișierul dumneavoastră HTML aceeași pereche de etichete <OBJECT></OBJECT> pe care o folosiți la prezentarea formularelor conștiente de date. Apoi, creați primul rând al tabelului folosind o pereche de etichete <THEAD></THEAD>. Imbricați capul de tabel într-o pereche de etichete <TABLE></TABLE>. Eticheta de deschidere <TABLE> necesită câteva atribute, precum în exemplul următor, în care un atribut ID dă tabelului numele mytable și un atribut DATASRC asociază obiectul sursă de date corespunzător cu tabelul:

```
<TABLE BORDER="1" ID="mytable" DATASRC="#mydata">
```

Atributul BORDER definește grosimea chenarului folosit între celulele din tabel.

Apoi creați trei linii după eticheta de deschidere <TABLE>, ca mai jos, care încep capul de tabel cu o etichetă <THEAD>, primul rând cu o etichetă <TR> și celula de antet a primei coloane cu o etichetă <TD>:

```
<THEAD>
<TR>
<TD><FONT COLOR="0000FF"><B><U><DIV ID=col1>Name</DIV></U></B></FONT></TD>
```

Dintre toate etichetele imbricate în perechea <TD></TD>, secvența <DIV ID=col1>Name</DIV> este cea mai critică. Diviziunea creează un obiect cu numele col1. Amintiți-vă că *diviziune* este doar un alt cuvânt pentru o secțiune, dar o diviziune este o secțiune care poate fi mult mai mică decât o secțiune de procesare de cuvânt tipică. Puteți face trimitere la col1 din oricare funcție script pentru a modifica aspectul sau conținutul diviziunii.

Textul Name apare în celula de tabel pe pagina Web. Ar trebui să recunoașteți etichetele <FONT>, <B> și <U> ca mai vechile etichete HTML care afectează aspectul textului. În acest caz, titlurile tabelului sunt colorate în albastru și sunt transformate în aldine subliniate. Recomand adăugarea vreunei forme de formatare a textului rândului de antet, pentru a-l face să se detașeze de restul tabelului.

OBSERVAȚIE

În Capitolul 29, ați învățat să variați formatul sau stilul textului de o manieră care să ofere utilizatorului o reacție inversă. Anteturile de tabel sunt un bun exemplu despre locul unde ar trebui să variați atributele textului astfel încât să atragă atenția către posibilitatea lor de a interacționa. Dacă adăugați sortare dinamică tabelului dumneavoastră, asigurați-vă că evidențiați anteturile. Utilizatorii sunt mult mai apți de a executa clic pe un antet dacă acesta le sare în ochi.

Pentru fiecare coloană din sursa dumneavoastră de date, adăugați o pereche de etichete <TD></TD>, similar exemplului anterior. După ce declarați toate capetele de coloană, încheiați primul rând de tabel cu o etichetă </TR> și închideți capul de tabel cu o etichetă </THEAD>.

După capul de tabel, creați un identificator de corp de tabel folosind perechea de etichete <TBODY></TBODY>. Într-un tabel cu patru coloane, identificatorul corpului de tabel ar putea să arate ca următorul:

```
<TBODY>
<TR>
<TD><DIV DATAFLD="Col1"></DIV></TD>
<TD><DIV DATAFLD="Col2"></DIV></TD>
<TD><DIV DATAFLD="Col3"></DIV></TD>
<TD><DIV DATAFLD="Col4"></DIV></TD>
</TR>
</TBODY>
```

Fiecare celulă din identificatorul corpului tabelului conține o diviziune cu un atribut DATAFLD. Valoarea atributului DATAFLD este numele de coloană specificat în sursa de date pe care ați precizat-o în eticheta de deschidere <TABLE>. În acest caz, folosiți sursa de date mydata.

OBSERVAȚIE

Ca a o reamintire, nu uitați că identificatorul sursei de date mydata este numele atributului ID al obiectului pentru perechea de etichete <OBJECT></OBJECT>. Adăugați paginii perechea de etichete pentru a furniza controlul ActiveX respectiv pe care browserul Web îl folosește pentru a trata funcționalitatea conștientă de date dinamică a paginii Web.

Asta-i tot ce aveți de făcut pentru a adăuga tabele cu expandare dinamică unei pagini Web care urmează abordarea Microsoft pentru conștientizarea datelor. Browserul Web face restul.

Acest capitol vă oferă două exemple de cum să generați dinamic tabele pornind de la o sursă de date. Primul demonstrează flexibilitatea unei pagini Web dinamice în tratarea surselor de date cu lungime variabilă. Al doilea exemplu arată cum să oferiți caracteristica de sortare dinamică cititorilor dumneavoastră.

Un exemplu de generare dinamică de tabel

Așa cum am menționat anterior, obțineți mai multe avantaje din generarea dinamică a unui tabel când prevedeți o sursă de date de lungime variabilă într-o perioadă de timp. O pagină Web „Raport asupra zăpezilor proaspăt căzute" este un exemplu, iar Listingul 30.5 vă oferă un altul.

Pentru acest prim exemplu, imaginați-vă că oferiți un sit Web care anunță nou-născuții dintr-un spital orășenesc aglomerat. Oricare cetățean al orașului poate vizita pagina dumneavoastră zilnic pentru a vizualiza informații despre copiii care au fost asistați de personalul spitalului. Trebuie să furnizați numele mamei, numele copilului (dacă există), sexul, greutatea și lungimea copilului și comentariile mamei (dacă există), toate într-un tabel dinamic pe pagina Web.

Priviți Listingul 30.5 și Figura 30.4, iar apoi continuați cititul pentru explicații detaliate.

Listingul 30.5. Pagina Web a nașterilor la spital.

```
<HTML>
<HEAD><TITLE>Recent Hospital Births</TITLE></HEAD>
<BODY BGCOLOR= "#FFFFFF">
<H2>This Month's Births</H2>
<HR>
<P>
<OBJECT ID="babylist"
  CLASSID="clsid:333C7BC4-460F-11D0-BC04-0080C7055A83"
  ALIGN="baseline" BORDER="0" WIDTH="0" HEIGHT="0">
```



```
  <PARAM NAME="DataURL" VALUE="babies1.txt">
  <PARAM NAME="UseHeader" VALUE="True">
</OBJECT>
<TABLE BORDER="1" ID="elemtbl" DATASRC="#babylist">
<THEAD>
<TR>
<TD><B><U><DIV ID=mom>Mom</DIV></U></B></TD>
<TD><B><U><DIV ID=name>Baby</DIV></U></B></TD>
<TD><B><U><DIV ID=sex>Sex</DIV></U></B></TD>
<TD><B><U><DIV ID=weight>Weight</DIV></U></B></TD>
<TD><B><U><DIV ID=length>Length</DIV></U></B></TD>
<TD><B><U><DIV ID=comment>Comment</DIV></U></B></TD>
</TR>
</THEAD>
<TBODY>
<TR>
<TD><DIV DATAFLD="Mom"></DIV></TD>
<TD><SPAN DATAFLD="Baby"></SPAN></TD>
<TD><DIV DATAFLD="Sex"></DIV></TD>
<TD><DIV DATAFLD="Weight"></DIV></TD>
<TD><DIV DATAFLD="Length"></DIV></TD>
<TD><DIV DATAFLD="Comment"></DIV></TD>
</TR>
</TBODY></TABLE>
<HR>
</BODY>
</HTML>
```

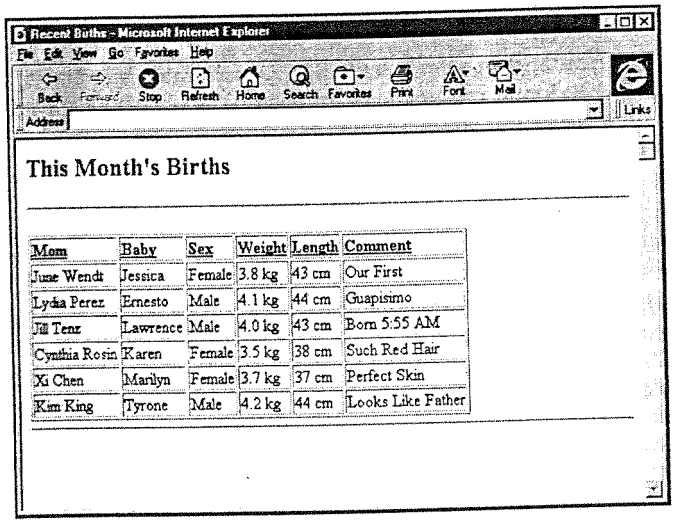


FIGURA 30.4.
Nașterile în trei zile din mai.

Înainte să revedeți Listingul 30.5 în detaliu, concentrați-vă pe beneficiile utilizării tabelelor generate dinamic. Figura 30.4 arată pagina Web This Month's Births, care afișează sursa de date din Listingul 30.6. După rândul de antet al coloanei, tabelul se expandează dinamic pentru a include șase rânduri în plus, fiecare cu o înregistrare de copil separată.

Listingul 30.6. Conținutul fișierului babies1.txt.

```
Mom,Baby,Sex,Weight,Length,Comment
June Wendt,Jessica,Female,3.8 kg,43 cm,Our First
Lydia Perez,Ernesto,Male,4.1 kg,44 cm,Guapisimo
Jill Tenz,Lawrence,Male,4.0 kg,43 cm,Born 5:55 AM
Cynthia Rosin,Karen,Female,3.5 kg,38 cm,Such Red Hair
Xi Chen,Marilyn,Female,3.7 kg,37 cm,Perfect Skin
Kim King,Tyrone,Male,4.2 kg,44 cm,Looks Like Father
```

Editați fișierul `babies1.txt` pentru a include două nașteri noi în data de 4 mai. Puteți să modificați fișierul, să îl salvați și apoi să reîncărcați pagina Web. Dacă sunteți în pană de inspirație, atunci doar copiați și alipiți sau tastați cele două rânduri de jos ale fișierului `babies2.txt` din Listingul 30.7. Tabelul va genera automat două rânduri noi, expandându-se pentru noua sursă de date. Amintiți-vă că oricare proces de calculator poate genera automat sursa de date cu zilele de naștere ale copiilor prin accesarea unei baze de date din rețeaua spitalului și executând o interogare. De asemenea, păstrați în minte că nu aveți nevoie să recreați pagina Web de fiecare dată când sursa de date se modifică.

Listingul 30.7. Conținutul fișierului babies2.txt.

```
Mom,Baby,Sex,Weight,Length,Comment
June Wendt,Jessica,Female,3.8 kg,43 cm,Our First
Lydia Perez,Ernesto,Male,4.1 kg,44 cm,Guapisimo
Jill Tenz,Lawrence,Male,4.0 kg,43 cm,Born 5:55 AM
Cynthia Rosin,Karen,Female,3.5 kg,38 cm,Such Red Hair
Xi Chen,Marilyn,Female,3.7 kg,37 cm,Perfect Skin
Kim King,Tyrone,Male,4.2 kg,44 cm,Looks Like Father
Jen Georgeson,Lyle,Male,4.1 kg,42 cm,First on Fourth
Tin Xin,Yu,Male,3.9 kg,42 cm,Very Easy Birth
```

Rezultatul adăugării a două noi linii la sursa de date este arătat în Figura 30.5. Datele din Listingul 30.7 sunt prezentate în browserul Web Internet Explorer 4.0. De această dată, tabelul afișează opt rânduri pentru opt înregistrări de copii din sursa de date plus unul pentru anteturile de coloană.

Îndreptați-vă din nou atenția asupra Listingului 30.5. Ați observat, probabil, că Listingul 30.5 nu reclamă un script. De fapt, nici nu ați avut nevoie să folosiți funcția de tratare a evenimentului `onLoad()` pentru a porni procesul de generare a tabelului. Browserul lansează tabelul și procesul de generare numai pe baza etichetelor HTML de pe pagina Web.

În primul rând, adăugați etichetele de obiect spre a activa controalele ActiveX care tratează generarea și expansiunea de tabele. Apoi, numiți obiectul cu atributul `ID="babylist"`. Vă veți referi la numele `babylist` ori de câte ori doriți să folosiți obiectul sursă de date în celelalte etichete HTML de pe pagină. Atributul `CLASSID` definește valoarea exactă pentru control. Nu aveți nevoie de spațiu fizic pe pagina Web pentru controlul de conștientizare a datelor, deci stabiliți atributele `BORDER`, `WIDTH` și `HEIGHT` la `0`. Adăugați un atribut `ALIGN` și dați-i valoarea `baseline`, care aliniază fiecare rând cu linia de bază a tabelului.

Cele două etichete `<PARAM>` sunt cerute de controlul ActiveX. Primul parametru pe care controlul ActiveX îl așteaptă este un `DATAURL` pentru sursa de date. Datele sunt salvate într-un fișier cu valorile separate prin virgule (CSV), numit `babies1.txt`. Doriți să folosiți primul rând din fișierul de date ca o înregistrare de cap de tabel (antet), deci stabiliți parametrul `UseHeader` la `True`.

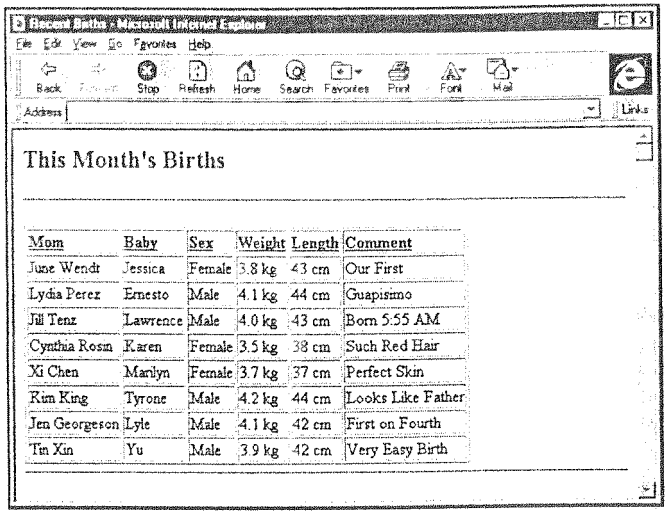


FIGURA 30.5. *Nașterile în patru zile din mai.*

Când sunteți gata, obiectul sursă de date arată astfel:

```
<OBJECT ID="babylist"
    CLASSID="clsid:333C7BC4-460F-11D0-BC04-0080C7055A83"
    ALIGN="baseline" BORDER="0" WIDTH="0" HEIGHT="0">
    <PARAM NAME="DataURL" VALUE="babies1.txt">
    <PARAM NAME="UseHeader" VALUE="True">
</OBJECT>
```

După ce definiți obiectul sursă de date, referiți-vă la valoarea atributului `ID` al sursei de date din tabel. Eticheta de deschidere `<TABLE>` include un atribut `DATASRC`, care s-a stabilit să fie numele obiectului sursă de date, `babylist`. Simbolul de început `#` este cerut de sintaxă atunci când vă referiți la un nume de sursă de date.

Apoi, creați primele două rânduri ale tabelului. Primul este imbricat în perechea de etichete `<THEAD></THEAD>`. Primul rând este un rând de anteturi de coloană pentru tabel. Creați primul rând cu etichete simple HTML, iar al doilea rând îl imbricați în perechea de etichete `<TBODY></TBODY>`. Al doilea rând este la fel de curat, dar include un atribut `DATAFLD` pentru fiecare etichetă de deschidere `<DIV>`. Valoarea atributului `DATAFLD` pentru fiecare diviziune este în concordanță cu rândul de antet din sursa de date. În caz contrar, pagina Web nu va prezenta datele corect în coloană, atunci când tabelul se expandează.

Un exemplu de sortare dinamică de tabel

Ca exemplu de sortare dinamică de tabel, această secțiune prezintă o pagină Web de statistici din baseball care prevede sortare dinamică de tabel. Puteți extinde această pagină astfel încât să includă statistici mai interesante, cum ar fi evoluțiile echipei acasă, numărul bazelor câștigate, numărul victoriilor la zero și procentajul de victorii acasă ale echipei.

Aruncați-vă privirea peste Listingul 30.8, apoi continuați să citiți pentru detalii. Cele mai interesante pentru dumneavoastră sunt funcțiile script de la sfârșitul listingului.

Listingul 30.8. O pagină de statistici din baseball care permit sortarea.

```
<HTML>
<HEAD><TITLE>WBL Current Standings</TITLE></HEAD>
<BODY BGCOLOR= "#FFFFFF">
<H2>WBL Current Standings</H2>
<HR>
<P>
<OBJECT ID="teamlist"
  CLASSID="clsid:333C7BC4-460F-11D0-BC04-0080C7055A83"
  ALIGN="baseline" BORDER="0" WIDTH="0" HEIGHT="0">
  <PARAM NAME="DataURL" VALUE="wblteam.txt">
  <PARAM NAME="UseHeader" VALUE="True">
</OBJECT>

This Table is Sortable By Clicking on the Column Header<P>
<TABLE BORDER="1" ID="elemtbl" DATASRC="#teamlist">
<THEAD>
<TR>
<TD><FONT COLOR="#0000FF"><B><U><DIV
  ID=team>Team</DIV></U></B></FONT></TD>
<TD><FONT COLOR="#0000FF"><B><U><DIV
  ID=wins>Wins</DIV></U></B></FONT></TD>
<TD><FONT COLOR="#0000FF"><B><U><DIV
  ID=losses>Losses</DIV></U></B></FONT></TD>
<TD><FONT COLOR="#0000FF"><B><U><DIV ID=BA>
  Batting Avg</DIV></U></B></FONT></TD>
<TD><FONT COLOR="#0000FF"><B><U><DIV ID=ERA>
  ERA</DIV></U></B></FONT></TD>
</TR>
</THEAD>
<TBODY>
<TR>
<TD><SPAN DATAFLD="Team"></SPAN></TD>
<TD><DIV DATAFLD="Wins"></DIV></TD>
<TD><SPAN DATAFLD="Losses"></SPAN></TD>
<TD><DIV DATAFLD="BA"></DIV></TD>
<TD><DIV DATAFLD="ERA"></DIV></TD>
</TR>
</TBODY></TABLE>
<SCRIPT LANGUAGE="JavaScript">
function teamClick() {
  teamlist.Sort = "Team";
  teamlist.Reset();
}

team.onclick = teamClick;

function winsClick() {
  teamlist.Sort = "Wins";
  teamlist.Reset();
}

wins.onclick = winsClick;

function lossesClick() {
  teamlist.Sort = "Losses";
  teamlist.Reset();
}

losses.onclick = lossesClick;

function baClick() {
  teamlist.Sort = "BA";
  teamlist.Reset();
}

ba.onclick = baClick;

function eraClick() {
  teamlist.Sort = "ERA";
  teamlist.Reset();
}

era.onclick = eraClick;

</SCRIPT>
<HR>
</FONT>
</BODY>
</HTML>
```


În afara funcțiilor din interiorul perechii de etichete `<SCRIPT></SCRIPT>`, Listingul 30.8 este foarte asemănător Listingului 30.5. Sursa de date folosită în acest exemplu este același fișier `wblteam.txt` folosit pentru exemplul cu vestiare dat anterior în Listingul 30.1.

Mai jos, sunt modificările pe care este nevoie să le faceți în Listingul 30.5 pentru a crea Listingul 30.8:

- Modificați atributul `ID` al obiectului sursă de date în `teamlist` și parametrul `DataURL` în `wblteam.txt`.
- Modificați oricare atribut `DATASRC` din `#babylist` în `#teamlist`.
- Modificați textul din rândul cap de tabel (s-a adăugat și culoarea albastră pentru font).
- Modificați oricare atribute `DATAFLD` în valorile corespunzătoare furnizate în linia de antet a sursei de date.

A fost adăugată o funcție script pentru a trata clicul de mouse al cititorului pe fiecare celulă de cap de tabel. Toate funcțiile au fost plasate în interiorul aceleeași perechi de etichete `<SCRIPT></SCRIPT>`. Fiecare funcție adaugă sufixul `Click()` la valoarea `ID` a etichetei `<DIV>` de deschidere a capului de tabel pentru a defini un nume de funcție. De exemplu, funcția pentru coloana `wins` este numită `winsClick()`.

Fiecare funcție conține două linii simple. Prima linie comunică browserului Web să sorteze din nou tabelul bazându-se pe coloana pe care cititorul execută clic cu mouse-ul:

```
teamlist.Sort = "Wins";,
```

Toate sortările sunt făcute în ordine crescătoare. Linia a doua comunică browserului Web să regenereze și să reexpandeze tabelul, ceea ce se face cu noua sortare:

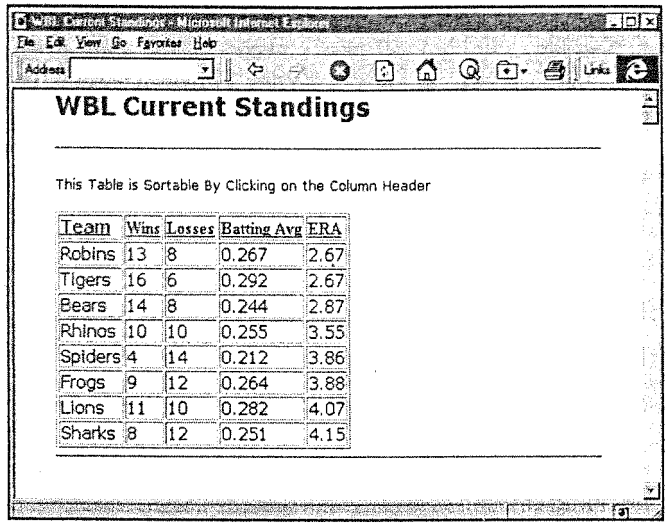
```
teamlist.Reset(),
```

OBSERVAȚIE

Este drept, nu se fac trimiteri la evenimentele funcției script de sortare ca atribute ale etichetei `<DIV>` pentru rândul de tabel, în schimb ele sunt asociate automat prin intermediul valorii atributului `ID`. Cred că este un mod inteligent de asociere a unui element cu un eveniment, dar el pune o povară mai mare pe browser, de a observa implicit conexiunea. În scop de documentare, îmi place să adaug atributul de eveniment elementului HTML care pornește evenimentul. Dumneavoastră puteți decide dacă vă place.

Adăugarea sortării dinamice la paginile dumneavoastră este simplă atunci când asocierea datelor se face corect. Ați putea să doriți să arătați cititorilor dumneavoastră că pot sorta tabelul în mod dinamic. În Figura 30.6, paginii i-a fost adăugat text pentru a arăta că tabelul poate fi sortat. Am sortat tabelul pe care îl vedeți în Figura 30.6 după raportul victoriilor (coeficientul ERA) executând clic pe titlul coloanei ERA.

FIGURA 30.6.
Sortarea clasamentelor dinamic.



Team	Wins	Losses	Batting Avg	ERA
Robins	13	8	0.267	2.67
Tigers	16	6	0.292	2.67
Bears	14	8	0.244	2.87
Rhinos	10	10	0.255	3.55
Spiders	4	14	0.212	3.86
Frogs	9	12	0.264	3.88
Lions	11	10	0.282	4.07
Sharks	8	12	0.251	4.15

DATAFORMATAS și DATAPAGESIZE

Puteți folosi atributul DATAFORMATAS în aceeași pereche de etichete ca a unui atribut DATAFLD pentru a face distincție între două tipuri diferite de date într-o bază de date: `text` și `html`. Dacă memorați etichetele HTML din textul unui câmp al bazei de date și creați un câmp de date pentru pagina Web cu un atribut `DATAFORMATAS=html`, atunci browserul Internet Explorer 4.0 formatează textul folosind etichete HTML. De fapt, puteți specifica chiar și o imagine prin memorarea unei perechi de etichete `<IMG></IMG>` în câmpul bazei dumneavoastră de date. Dacă nu identificați un atribut DATAFORMATAS, browserul folosește în mod prestabilit tipul text.

Ca un exemplu de folosire a atributului DATAFORMATAS, luați în considerare următoarea pereche de etichete `<DIV></DIV>`, care accesează dinamic o sursă de date:

```
<DIV DATAFLD="Element" DATAFORMATAS=html></DIV>
```

Ați putea avea un câmp de coloană `Element` care să arate ca acesta:

```
<B>Na</B> is the simbol for <I>Sodium</I>
```

Câmpul ar apărea pe pagina Web ca mai jos:

```
Na is the symbol for Sodium
```

Pe de altă parte, atributul DATAFORMATAS ar putea fi stabilit ca `text` sau să nu apară de loc, precum în acest exemplu:

```
<DIV DATAFLD="Element"></DIV>
```


În acest caz, câmpul ar apărea pe o pagină Web după cum urmează:

```
<B>Na</B> is the simbol for <I>Sodium</I>
```

În ultimul caz, etichetele apar, de fapt, ca text. Fără îndoială, doriți să evitați să aveți etichete care apar pe pagina dumneavoastră ca text. Încă o dată, capacitatea de a memora etichete HTML într-o bază de date și de a lăsa un browser Web să prezinte textul și grafica formatate este o caracteristică puternică. Din ce în ce mai mult, autorii de pagini Web vor stoca datele care se modifică într-o bază de date. De ce? Deoarece costul întreținerii paginilor Web scade semnificativ dacă subiectele modificărilor dinamice sunt păstrate separat de textul static.

Puteți folosi atributul DATAPAGESIZE pentru a preciza câte rânduri de date doriți să prezentați utilizatorului o dată. Pentru tabelele foarte lungi, utilizatorii ar putea dori să hotărască ei câte rânduri să vadă o dată. Puteți crea controalele care permit utilizatorilor să coordoneze prezentările de tabele.

Ca exemplu de folosire a atributului DATAPAGESIZE, considerați următoarea etichetă `<TABLE>` de deschidere a unui tabel dinamic:

```
<TABLE DATAPAGESIZE=10 id=Periodic DATASRC="#ElementTable" >
```

Acum, un utilizator va vedea numai zece înregistrări o dată când se generează tabelul. În scripturile dumneavoastră, vă puteți deplasa înainte folosind metoda `.nextPage` sau puteți modifica dinamic dimensiunea paginii folosind proprietatea `.dataPageSize` a obiectului nume de tabel.

Când utilizatorul execută clic pe un buton pe care îl prevedeți, evenimentul dumneavoastră `onmouseclick` poate executa o funcție script cu următoarea linie:

```
Periodic.nextPage
```

Puteți modifica dinamic dimensiunea tabelului la 15 înregistrări cu linia:

```
Periodic.dataPageSize=15
```

Atât DATAFORMATAS, cât și DATAPAGESIZE sunt atribute noi în HTML-ul dinamic, cu posibilități puternice.

Rezumat

Acest capitol a introdus conceptele de asociere a datelor și conștientizare a datelor. Asocierea datelor este o caracteristică a browserului Internet Explorer 4.0 care permite furnizarea de date flexibilă de către un autor de pagini Web. O dată ce o sursă de date este asociată unei pagini Web conștiente de date, aceasta ține cont de sursa de date descărcată în același timp cu pagina. Folosind HTML dinamic, puteți, atunci, să prevedeți un mod în care cititorul să interacționeze cu pagina Web și să filtreze sau să sorteze datele. Datele sunt, apoi, prezentate improspătate.

Avantajul folosirii conștientizării datelor include eliminarea cererilor multiple la serverul Web și eliminarea timpului de așteptare al cititorilor. Dezavantajul rezidă în timpul inițial de încărcare a sursei de date. Lumea corporațiilor accesează date prin rețea de ani de zile, iar specialiștii în servicii informatice (Information Services, IS) iau întotdeauna în considerare compromisurile în strategia de livrare a datelor când gândesc aplicații noi. Conștientizarea datelor paginii Web pune autorii de pagini Web în fața unor decizii similare privind compromisurile. Cu multiple strategii de distribuție disponibile, autorul de pagini web poate controla cel mai bun mod de a distribui datele din pagina Web către utilizator.

Cu Internet Explorer 4.0, puteți prezenta la un moment dat mai multe înregistrări ale unei surse de date dintr-o pagină Web. Afișarea doar a unui subset de înregistrări o dată este numită filtrare. Dacă doriți să prezentați întreaga sursă de date, puteți configura o generare și expandare automată de tabel folosind etichetele HTML dinamic adecvate. Tabelul se va umple automat cu date din fișierul sursă de date pe care l-ați precizat.

După ce o pagină Web încarcă automat o sursă de date într-un fișier, puteți indica după care coloană pot cititorii să sorteze tabelul. Cititorii sortează o coloană executând clic pe ea cu mouse-ul. Când se cere o sortare, tabelul se regenerează și se reexpandează dinamic, în ordinea liniilor din coloana sortată.

PARTEA a VII-a

Designul efectiv al unei pagini Web

Capitolul 31
Realizarea unei interfețe eficiente

de Molly E. Holzschlag

În acest capitol:

- **Principiile proiectării interfeței cu utilizatorul 460**
- **Analiza scopului unui sit 462**
- **Un exemplu concludent: Microsoft Site Builder Network 464**
- **Concepția proiectului 466**
- **Proiectarea interfeței – limbajul standard HTML 470**
- **Proiectarea interfețelor sofisticate 474**

După cum zîmbetul pe care-l afișați și aspectul exterior îi ajută pe ceilalți să identifice trăsăturile interesante și atrăgătoare ale persoanei dumneavoastră, așa și interfața cu utilizatorul este cea care le stârnește vizitatorilor dorința să se oprească și să savureze generozitatea oferită de un sit Web.

Interfețele îl întâmpină, îl îndrumă și îi furnizează vizitatorului dumneavoastră elementele funcționale necesare pentru a obține informațiile sau experiența căutată. Fiind considerată un aspect critic al proiectării multimedia, o interfață bine construită este în mod special importantă pentru Web. Dacă cineva nu găsește informația necesară sau dacă nu se bucură de o experiență semnificativă în paginile dumneavoastră, poate face o schimbare bruscă de drum și vizita alt sit – unul care este probabil mai interesant, mai informativ pentru vizitator și profitabil pentru competitorii dumneavoastră.

O metodă de evitare a realizării unor situri care funcționează ca o oprire la stand pe durata unei curse automobilistice sau ca o curbă „ac de păr" pe magistralele și drumurile mai puțin cunoscute ale sitului Web este de verifica dacă siturile pe care le construiți fac vizitatorul să se simtă confortabil și îi oferă bunurile pe care le caută. Sunt mai multe metode de construire a unor astfel de situri, precum proiectarea inteligentă și crearea unei interfețe atractive și *utile* cu situl.

În acest capitol vă voi prezenta principiile de bază ale proiectării interfeței și vă voi arăta cum să analizați scopul sitului, să concepeți un proiect care să țină cont de preocupările

publicului și să creați interfața corespunzătoare folosind procedeele HTML. Conceptele și procedeele din acest capitol sunt destinate să inspire creativitatea dumneavoastră și să fie imediat aplicate în proiectele curente.

Principiile proiectării interfeței cu utilizatorul

În cadrul planurilor pentru situl dumneavoastră pot fi aplicate mai multe principii de proiectare a interfeței cu utilizatorul, care și-au dovedit valabilitatea în timp. Aceste linii generale sunt extrase din alte produse media, ca de exemplu CD-ROM-urile, chioșcurile sau chiar televiziunea! Mediul hipermedia caracteristic unui sit Web – cu legăturile sale aici, acolo și pretutindeni în lumea vastă și complexă Internet – este adesea lipsit de aceste principii fundamentale. Rezultatul sunt interfețele prost proiectate, care provoacă confuzie și frustrare, mai degrabă decât să informeze și să asiste pe cei care vizitează siturile Web respective.

Motivul abundenței problemelor generate de interfața cu utilizatorul în rețea este legat de faptul că siturile Web sunt adeseori construite de ingineri de calculatoare, elevi de liceu sau chiar artiști talentați. Toți aceștia pot contribui mult în ceea ce privește conținutul mediului Web, dar au foarte puțină experiență în ceea ce este necesar pentru comunicarea în structura neobișnuită, neliniară a sitului Web.

SUGESTIE

Vă poate ajuta un „sărut" (KISS). Așa este, urmați acel acronim înțelept (chiar dacă nu este prea delicat spus), corespunzător expresiei Keep It Simple, Stupid! (Păstrează simplitatea, tâmpitule!). Interfețele cu utilizatorul pot fi complicate tehnologic, dar ele trebuie să fie ușor înțelese de oricare utilizator.

Următoarele principii vă pot ajuta să evitați gropile în care cad autorii de pagini Web neexperimentați. Aplicați aceste idei la situl pe care-l construiți și veți avea mai multe șanse de a călători liniștit către succesul Web.

Metafora

În proiectare, *metafora* se referă la reprezentarea simbolică a structurii pe care încercați să o construiți. O metaforă acționează ca un mijloc vizual familiar, în jurul căruia construiți căi de acces, interioare, ferestre, uși și ieșiri din mediul dumneavoastră.

De fapt, pentru a scrie paragraful anterior am folosit metafora. Am definit un sit Web ca și cum ar fi o clădire – cu o serie de elemente pe care vă așteptați să le găsiți într-o clădire. Metafora ajută oamenii să se simtă confortabil, deoarece sunt familiarizați cu configurația. Puteți găsi un bun exemplu de utilizare a metaforei în Microsoft Network, care folosește o metaforă tip TV sau radio când denumește domeniile sale „canale", după cum se vede în Figura 31.1. Utilizatorii știu ce este un canal, așa că ei pot face ușor legătura cu conceptul și pot interacționa cu interfața fără să se gândească prea mult *cum* să facă acest lucru.

Metafora trebuie să utilizeze concepte comune, cotidiene, pe care persoanele din oricare colț al lumii care intră în situl dumneavoastră le pot înțelege imediat. Realizați metafora și sunteți cu un pas mai aproape de a ajuta oamenii să se simtă confortabil și să rămână un timp cu dumneavoastră.

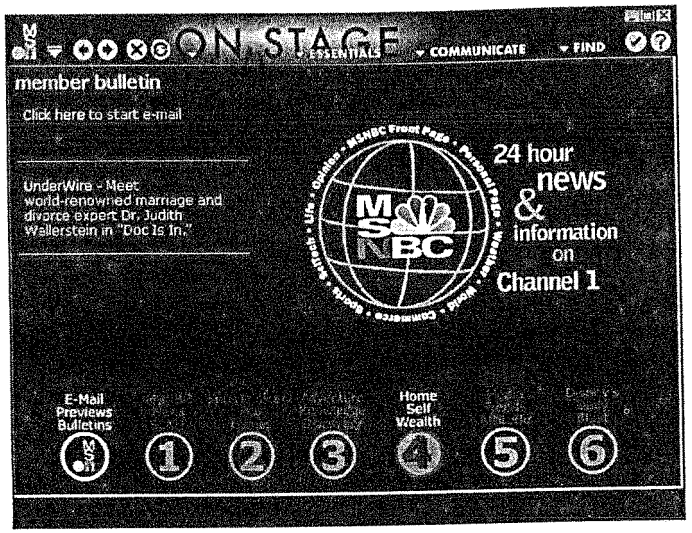


FIGURA 31.1. *Metafora corespunzătoare familiarului „canal" TV este folosită în Microsoft Network, versiunea 2.0 din 1997.*

Claritatea

Pentru a-i crește vizitatorului dorința de a rămâne, trebuie să fiți sigur că acesta înțelege elementele din paginile dumneavoastră. Nici una din piesele critice nu trebuie să fie abstractă sau greu de descifrat. Aceasta nu înseamnă că abstracția ca o formă de artă nu este permisă – un bun proiectant poate utiliza arta abstractă într-un sit Web foarte clar. Ceea ce un bun proiectant nu trebuie să facă este să folosească abstracțiile când este vorba despre elementele necesare pentru navigarea în sit, localizarea informației sau întoarcerea la alte domenii importante din sit.

Elementele care intră în această categorie cuprind butoanele, hărțile imagine sau legăturile necesare pentru navigarea în sit. Un buton care conduce vizitatorul spre stânga nu ar trebui să conțină o imagine cu o săgeată îndreptată în sus, iar o legătură care oferă o opțiune de poștă nu trebuie să oprească prezentatorul de informații (newsreader). Lucrurile trebuie să fie simple și clare. Claritatea este obligatorie pentru o comunicare precisă.

Consecvența

Consecvența este nu numai de o deosebită importanță în proiectarea unei interfețe, dar este și una din necesitățile de bază ale sitului Web. Prea des mi se întâmplă să aterizez pe o pagină Web și să gândesc: „Arată nemaipomenit!". Apoi, pe măsură ce trec la următoarea pagină, mă surprind mirându-mă ce s-a întâmplat cu aspectul tentant și promisiunea oferită de prima pagină. Dacă rămân destul pentru a explora situl, fondul se schimbă, stilul de fonturi este inconsecvent, antetul și navigarea sunt complet neregulate, pe scurt, nu pot spune, de la o pagină la alta, unde naiba sunt în lumea Web!

Consecvența în proiectarea elementelor permite coeziunea în prezentare. Un aspect unitar al sitului dumneavoastră îl păstrează pe vizitator calm, în loc să-l facă încordat, confuz și gata de a vira stânga sau dreapta din situl dumneavoastră.

Orientarea și navigarea

Urmând îndeaproape aceeași linie a conceptelor anterioare, este și ideea că vizitatorul sitului trebuie să cunoască unde este în orice moment. Acest principiu este *orientarea*. Dacă sunt adânc cufundat într-un sit care are sute de pagini, îmi este de folos să știu unde mă găsesc în sit. Este de asemenea util să am acces rapid la alte domenii ale sitului și, dacă constat că sunt undeva unde de fapt nu vreau sau nu am nevoie să fiu, să pot reveni în locul de unde am plecat.

Orientarea este realizată prin dotarea fiecărui sit cu un antet care definește scopul paginii sau cu alt element familiar care-mi spune instantaneu unde sunt. Ați văzut probabil o diversitate de metode folosite pentru a asigura orientarea. Un exemplu poate fi găsit în situl Wilde Rose Cofee Roastary, `http://desert.net/wilderose/`. Observați că o ceașcă de cafea goală mă ajută să-mi definesc poziția (vezi Figura 31.2). După cum puteți constata, navigarea este legată nu numai de orientare, dar și de celelalte premise ale interfeței cu utilizatorul, pe care vi le-am prezentat. Voi explica acest lucru mai detaliat în Capitolul 32, „Navigarea și machetarea".

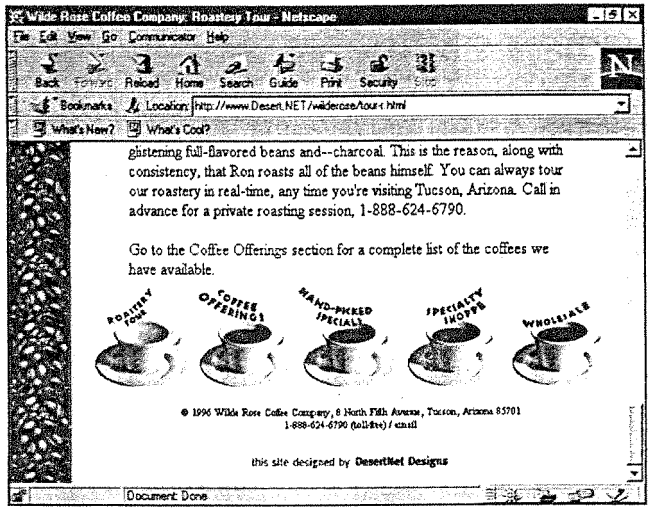


Figura 31.2. *Observați că ceașca de cafea goală corespunde cu orientarea mea față de pagina curentă.*

SUGESTIE

Numai unul, două sau trei clicuri! Dacă cereți vizitatorului să facă clic de mai mult de trei ori pentru a obține informația pe care o caută, riscați să-l dezorientați. Ideal, două informații corelate din situl dumneavoastră trebuie să fie la cel mult trei clicuri una de alta.

De acum încolo voi spune pur și simplu că navigarea este o parte integrală a proiectării interfeței și un element critic al oricărui sit Web. Totul este să ajungi de undeva exact în locul dorit – logic, rapid și ușor.

Analiza scopului unui sit

Ca și în cazul altor proiecte, unul din cei mai importanți pași este să analizați scopurile și să planificați atingerea acestora. Un sit Web bun este întotdeauna bine planificat, altfel rezultatul poate fi întâmplător și confuz.

OBSERVAȚIE

Dezvoltatorii care au lucrat în grupuri organizate sunt deja familiarizați cu analiza audienței și a scopului. Totuși, mulți proiectanți vin pe diferite căi în domeniul Web. Indiferent de tipul sitului, pagină a unei firme sau personală, principiile discutate aici sunt întotdeauna folositoare.

Un început bun este să detaliem scopul sitului și definiția publicului. Comparația vă permite să cunoașteți ce direcții să urmați în planificarea de scurtă și lungă durată, ce tipuri de tehnologii vă vor ajuta la proiectare și desigur, ce tipuri de interfață și de proiectare a sitului vor satisface cel mai bine nevoile dumneavoastră.

Scopul proiectului

Începeți prin a răspunde la următoarele întrebări:

- Care este motivul principal de a avea acest sit?

Urmează situl să fie un punct de vânzare? Sau poate sunteți interesat să furnizați servicii pentru client, sau asistență tehnică pentru produsele firmei dumneavoastră. Alte posibile scopuri ale siturilor Web includ distracțiile, publicitatea și promovarea produselor, informații și educație, știrile și comunitățile cu anumite preocupări.

- Ce informații curente și conținut aveți la dispoziție și considerați că merită să fie incluse în sit?

Broșuri de firmă, fotografii de produse și ilustrații, campanii de reclamă anterioare, programe interesante și jocuri, reproduceri de știri, artă, adresa URL a unui Web similar interesant și o gamă de alte materiale pe care deja le aveți vă vor ajuta să definiți conținutul sitului dumneavoastră.

- Care sunt scopurile sitului pe termen scurt și lung?

Cea mai bună cale de a aprecia scopurile pe termen scurt și mediu este de a cunoaște care este scopul imediat. Dacă situl dumneavoastră intenționează să ofere servicii de client pentru calculatoare portabile și există cerințe pentru furnizarea imediată a asistenței online pentru clienți, mai întâi stabiliți care modele de calculatoare portabile necesită cele mai multe servicii și asistență.

Apoi, încercați să priviți dincolo de cererile imediate, pentru a stabili o strategie pe termen lung. Această analiză vă va ajuta să determinați structura sitului. Veți începe prin a furniza ceea ce clienții dumneavoastră au nevoie astăzi, dar veți scăpa de foarte multe dureri de cap prin planificarea a ceea ce vor cere mâine.

Stabilirea publicului

Acum, definiți publicul:

- Care sunt caracteristicile demografice?

Încercați să vindeți cărți rare colecționarilor? Poate doriți să creați un sit care să strângă fonduri pentru scleroza multiplă. Ați putea avea un ziar care se adresează persoanelor educate cu vârsta de aproximativ 30 de ani sau poate doriți să creați un sit Web interactiv pentru membrii unei comunități homosexuale dintr-un anumit oraș. Indiferent de scopul dumneavoastră, *modul* de proiectare al interfeței va fi afectat de *cine* este destinatarul .

- Cu ce experiență sau informații trebuie să plece publicul?

Răspunsul la această întrebare este deosebit de important. Dacă ați analizat și planificat să satisfaceți toate cerințele, dar vizitatorul părăsește situl fără o anume experiență, activitate

sau resursă, ați ratat efectul. Conștient de faptul că doriți ca vizitatorul să părăsească situl înarmat cu cunoștințe privind opțiunile de planificare familială sau cu un itinerariu de vacanță disponibil, puteți să proiectați situl pentru a atinge acel scop.

O dată ce ați răspuns cu succes la întrebări, luați în considerare scopurile în context demografic. Acest pas este extrem de important, pentru că, în momentul în care vă așezați să proiectați și să scrieți codul pentru situl dumneavoastră, veți avea o idee clară asupra tipului de interfață ce trebuie proiectată! Dacă situl dumneavoastră este bine planificat și organizat, eforturile vor fi mai bine coordonate și mai eficiente.

Examinând informația pe care ați cules-o, veți descoperi că multe lucruri importante sunt cunoscute. În primul rând, cunoașteți scopul sitului și obiectivele pe termen lung. Această cunoaștere este legată de proiectarea practică a interfeței, deoarece vă spune de ce aveți nevoie pentru proiectare și cât de mult trebuie proiectat.

De asemenea, știți ce informații curente sunt disponibile pentru conținut și cine formează audiența, lucruri care determină tipul de interfață pe care-l veți folosi pentru a-i transmite acea informație. În fine, cunoașteți ce doriți să câștige audiența din situl dumneavoastră Web. Sunteți acum gata să priviți, să pipăiți și să mirosiți proiectul, să-i vedeți limitele și conținutul pagină cu pagină.

Un exemplu concludent: Microsoft Site Builder Network

Un bun exemplu pentru rezultatul definirii audienței și a intenției sitului poate fi găsit în cazul Site Builder Network al companiei Microsoft. Acest sit activ, în continuă creștere, oferă informații, activități comunitare (precum avizierele electronice și discuții) și instruire pentru autorii de pagini Web. Vizitați situl mutând browserul Web la `http://www.microsoft.com/sitebuilder/`.

În acest caz, scopul sitului este de a furniza informații și servicii către persoane asemănătoare dumneavoastră, care ar putea proiecta situri pentru Internet sau pentru rețelele intranet sau extranet ale companiei dumneavoastră. Există scopuri pe termen scurt, ca de exemplu oferirea celor mai noi știri din domeniul industrial (Figura 31.3) și pe termen lung, ca de exemplu construirea de resurse durabile și al unui sentiment de comunitate, care sunt furnizate de serviciile de grup de informare și de reacții. Prin adresarea acestor elemente se poate realiza o interfață clară, deoarece un scop clar are ca rezultat o transmitere mai clară a acestuia.

OBSERVAȚIE

Internet, intranet și extranet, vai de mine! Care-i diferența? Simplu spus, siturile Web Internet sunt cele găsite în zonele comune de trafic ale rețelei. Ele sunt situri Web accesibile oricărei persoane cu conexiune Internet și browser Web. Intraneturile sunt rețele protejate de parafocuri (*firewalls*), care păstrează informația în interiorul unei anumite zone. Rețelele intranet sunt folosite pentru a furniza informații firmelor și organizațiilor de la terminale și calculatoare folosite de angajați. Extranetul este poziționat în spațiul virtual dintre Internet și intranet. Este o parte a intranetului care este deschisă traficului mai larg, dar totuși controlat. În cazul unei edituri, extranetul ar putea fi o rețea destinată cumpărătorilor de cărți, celorlalte edituri și persoanelor care lucrează în domeniul respectiv.

Conținutul oferit de Microsoft este adeseori generat de o mulțime de autori și de proiectanți din cadrul firmei. Totuși, în tot situl sunt disponibile legături ample la resurse externe, care demonstrează spiritul comunității și oferă soluții alternative celor care sunt probabil nevoiți să proiecteze pentru alte platforme decât cele acceptate de Microsoft.

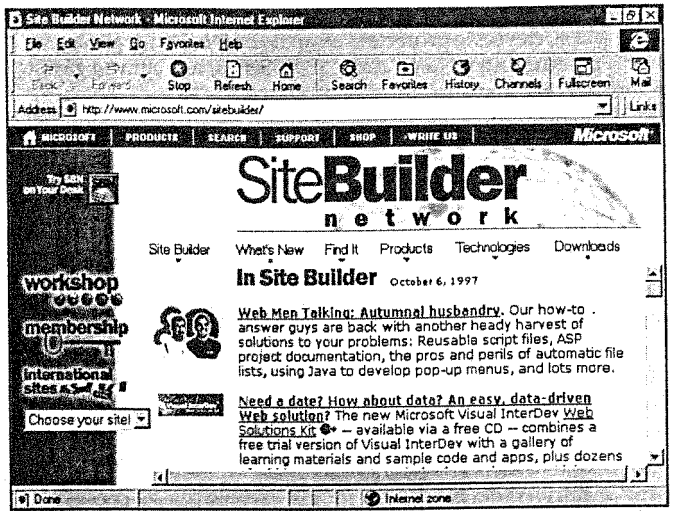


FIGURA 31.3. *Pagina „What's new" (Ce este nou) a rețelei Microsoft Site Builder Network.*

Audiența sitului este foarte specifică, aceasta fiind caracteristica definitorie a succesului proiectării interfeței cu situl. Deoarece audiența este constituită din proiectanți, concepția poate fi flexibilă și folosește cadre și tehnologii de programare, ca de exemplu Visual Basic Script (VB Script), pentru a furniza produsele (Vezi Figura 31.4). Motivul pentru care situl poate utiliza aceste procedee este că un autor de pagini Web este mai mult ca sigur dotat cu browserele cele mai recente și cu hardware, programe plug-in și servicii de vârf.

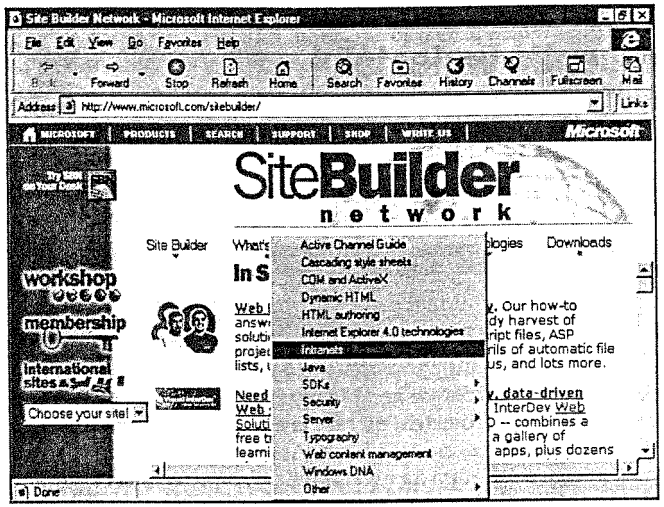


FIGURA 31.4. *Interfața cu Site Builder Network este construită cu tabele, cadre și meniuri autoderulante folosind VBScript.*

Pentru a ilustra rolul jucat de public în modul în care Microsoft a decis să proiecteze interfața cu situl, priviți Figura 31.5. Acesta este o fotografie a ecranului paginii principale a companiei Microsoft, la adresa `http://www.microsoft.com/`. Publicul acestei pagini este format din utilizatori de calculatoare despre care nu se știe nimic altceva decât că sunt

interesați de produsele Microsoft. Acest lucru este destul de general. Așa că interfața este mai simplă și, deși este proiectată atractiv, cu tabele și grafică, se evită folosirea tehnologiei de vârf, precum interfața cu cadre și VBScript, așa cum se găsește în pagina principală a Site Builder Network.

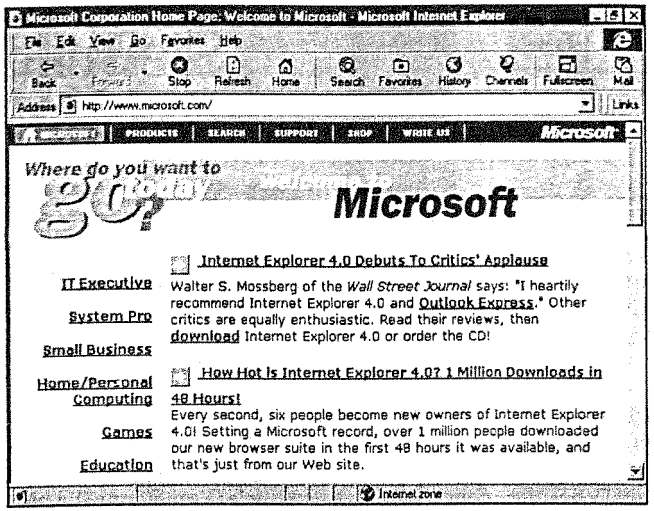


FIGURA 31.5.
Deși pagina de bază a Microsoft folosește tabelele, interfața evită folosirea cadrelor sau scripturile datorită bazei largi a publicului.

În sfârșit, informația este nu numai bine organizată și accesibilă, dar și bogată. Vizitatorii acestui sit care caută informații privind tehnologia de vârf, HTML, ActiveX, Java, le vor găsi și vor părăsi situl cu nenumărate resurse.

Concepția proiectului

O dată ce ați înțeles în ce constă proiectarea interfeței, ce se cere de la dumneavoastră pentru a analiza scopul sitului și publicul și sunteți înarmat cu un exemplu de interfață de succes, sunteți gata să începeți să gândiți cum să implementați un proiect.

Modul efectiv de exprimare a cunoștințelor câștigate în domeniul tehnologiei Web depinde în mare măsură de abilitățile și resursele umane și tehnologice de care dispuneți. Nu orice persoană care citește acest capitol va avea abilități HTML sau de proiectare avansate, dar unii vor avea. O altă problemă este întinderea materialului pe care doriți să-l furnizați. Un cititor ar putea dori numai să promoveze serviciile sale profesionale ca terapeut maseur, dar altcineva care citește materialul din această carte caută informații despre crearea interfețelor cu facilități de răspuns și conversație pentru un cotidian.

Pentru a lua în considerare aceste circumstanțe diferite și a vă furniza informații care să vă ajute să concepeți proiectele pe care le aveți în minte, voi parcurge mai întâi unele dintre elementele sitului cu care veți lucra. Apoi voi prezenta două abordări practice ale proiectării interfeței: o interfață de nivel standard, care necesită cunoștințe HTML și de grafică de bază, și o interfață avansată, care necesită tabele pentru machetă.

OBSERVAȚIE

În Capitolul 32, vă voi prezenta exemple chiar mai extinse și mai sofisticate de interfețe.

Pagini obișnuite într-un sit Web

Iată o listă a paginilor obișnuite într-un sit Web:

- Pagina de întâmpinare;
- Pagina de cuprins;
- Pagina de reacție;

Alte aspecte cu care s-ar putea să fie nevoie să vă familiarizați sunt următoarele:

- Mijloace de informare descărcabile (programe, fișiere, sunete și video);
- Mijloace de informare inline (audio, video și prezentări multimedia care rulează în cadrul parametrilor browserului);
- Funcții de căutare;
- Camere de discuții, aviziere electronice și grupuri de informare.

Tendința actuală este să se includă elementele comune și aspectele specifice într-un singur format integrat. O pagină de întâmpinare poate diferi foarte mult față de o pagină plină de conținut, în ceea ce privește aspectul și impresia pe care o produce, iar paginile de conținut sunt diferite de cele necesare pentru informarea inline, cum este de exemplu o prezentare Surround Video a noii mașini pe care o lansează compania dumneavoastră.

Utilizând noțiunile de bază pe care le-ați învățat, precum consecvența, claritatea, orientarea și navigarea bine concepută, ar trebui să puteți păstra destul de ușor unitatea conceptuală de la o pagină la alta, chiar în cazul unui sit foarte mare.

Intrarea

Prima parte a proiectării interfeței o reprezintă luarea deciziei privind modul cum va arăta intrarea în situl dumneavoastră și cum se va integra aceasta cu paginile de conținut ale sitului. Unii proiectanți preferă o pagină preponderent grafică, foarte asemănătoare cu coperta tradițională a revistelor. Alți proiectanți preferă o întâmpinare cu caracter funcțional sau *pagina de deschidere (splash page)*, cu grafică, dar și cu opțiuni de navigare, iar alții sunt de părere că pagina de deschidere este o pierdere de timp, deoarece oamenii vor să ajungă rapid la informație.

Fiecare client sau sit va avea cerințe unice, de aceea vă sfătuiesc să folosiți cercetările făcute privind scopurile și publicul pentru a determina cea mai potrivită soluție în condițiile date. Unii proiectanți preferă o pagină de deschidere, alții nu. Indiferent de soluția aleasă, consecvența în proiectare este un element critic.

ATENȚIE!

Liantul unui sit este integrarea, așa că, dacă aveți o pagină de deschidere care nu seamănă deloc cu paginile din interior, nu aveți interfață. Asigurați un aspect uniform al paginilor dumneavoastră.

O pagină de deschidere corespunzătoare trebuie să reprezinte identitatea sitului, dar și să introducă unele elemente de proiectare ale acestuia, ca de exemplu culoarea, forma, tipografia și textura (vezi Figura 31.6).

Dacă nu doriți o pagină de deschidere, asigurați-vă de faptul că vizitatorul este binevenit în pagina dumneavoastră de bază. Chiar dacă aveți informații suficiente și un sistem de navigație detaliat pe prima pagină, verificați dacă vizitatorul știe cine sunteți și de ce sunteți acolo. Este de asemenea bine să spuneți sau să sugerați vizitatorului ce sperați să parcurgă sau să câștige din vizită.

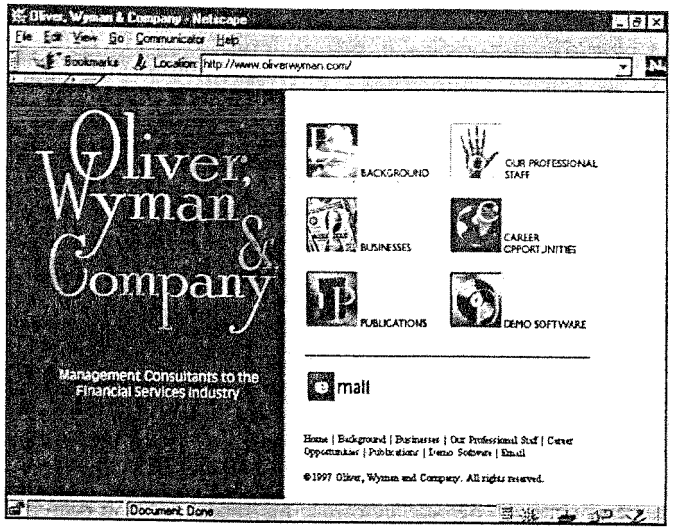


FIGURA 31.6. *Pagina de deschidere Oliver Wyman reprezintă identitatea sitului și oferă o introducere a elementelor interne de proiectare.*

OBSERVAȚIE

Există excepții de la regulă ca „întâmpinarea este obligatorie" sau „Este necesară prezentarea pe o pagină de deschidere" ? Da, dar acestea sunt determinate de publicul sau de scopul dumneavoastră și excepțiile apar mai des la siturile pentru distracții, cele pentru jocuri sau la siturile artistice decât la siturile de firmă, pentru consumatori sau pentru asistența clienților.

Țineți minte, intrarea este prima dumneavoastră ocazie de a face oamenii să se simtă bineveniți și interesați de conținutul care va urma. Gândiți-vă bine și aplicați principiile de proiectare a interfeței cu utilizatorul pentru a obține o relație imediată cu vizitatorii dumneavoastră!

Paginile de conținut

Pe măsură ce vizitatorul se deplasează într-un sit, fiecare pagină ar trebui să ofere o combinație de consecvență și caracteristici noi. Consecvența poate fi asigurată de particularități precum paletele de culori și fonturile, iar componente noi pot fi adăugate folosind o diversitate de machete, grafică și alte opțiuni multimedia.

În Figura 31.6 puteți vedea situl Oliver Wyman. Se prezintă paleta de culori, precum și metafore cum ar fi pictogramele ziar, pentru știri, respectiv pictograma cer în fundal, pentru informații generale privind compania.

Obiectivul interfeței este de a menține interesantă fiecare pagină, astfel încât vizitatorii să simtă impulsul de a vedea ce urmează, nu numai pentru informația pe care trebuie s-o dezvăluiți, dar și pentru panorama vizuală care se desfășoară când aceștia explorează situl. Înainte ca vizitatorul să intre într-o pagină interioară a sitului Wyman, i s-au prezentat deja trăsăturile specifice. Pe acest fundament, proiectantul poate adăuga noi elemente de machetare sau de proiectare. Adaosul progresiv de informații protejează consecvența și previne dezorientarea vizitatorului (vezi Figura 31.7).

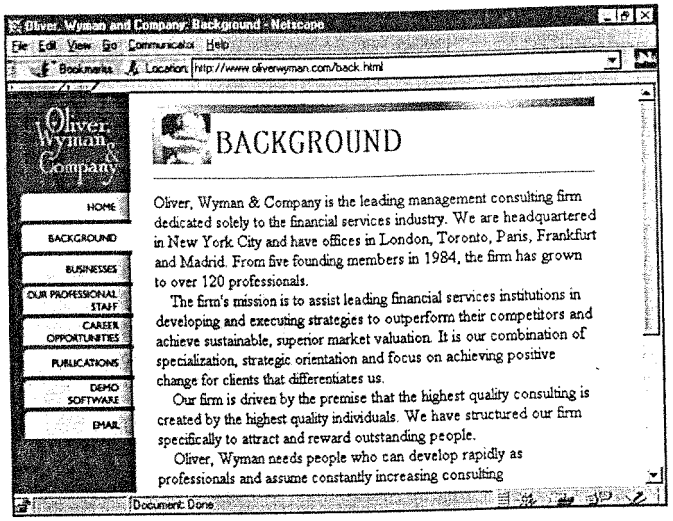


FIGURA 31.7. *Paginile cu conținut ale sitului Oliver Wyman păstrează aspectul consecvent, dar interesant, astfel încât atrag atenția vizitatorilor.*

Formularele de reacții

Și formularele de reacții trebuie să fie considerate ca parte a proiectului. Mulți proiectanți talentați păstrează consecvența aspectului sitului, dar o pierd brusc când nu concep un formular de reacții clar, atractiv, care să reflecte concepția sitului.

Păstrați aceleași culori, fonturi și machete când creați formularele. Unde este posibil, aliniați marginea din dreapta a câmpurilor de răspuns. Această aliniere determină un aspect curat și ordonat, ceea ce îmbunătățește claritatea. S-ar putea ca vizitatorii Web să nu-și dea seama de ce formularele dumneavoastră sunt așa de ușor de utilizat, dar vor aprecia acest lucru și-și vor aminti de situl dumneavoastră ca de unul confortabil, pe care îl vor utiliza din nou și chiar îl vor recomanda prietenilor interesați.

Considerații speciale

Cele trei elemente acoperite până acum – paginile de întâmpinare, paginile de conținut și paginile de răspuns – sunt cele mai folosite pe siturile Web. Totuși, se manifestă o cerere crescândă atât din partea industriei competitive cât și din partea publicului interesat pentru furnizarea unui conținut cu interactivitate ridicată, care să folosească scriptul, bazele de date, multimedia și procedee de programare Web avansate.

În Figura 31.8 puteți vedea un exemplu de prezentare multimedia a sitului Web al Muzeului Deșertului Arizona-Sonora (`http://desert.net/museum/`), care necesită o porțiune destul de mare din pagină. De aceea, proiectul interfeței trebuie armonizat cu mijlocul de informare. Cum păstrați unitatea în proiectarea interfeței când un obiect folosește atât de mult spațiu?

Dacă v-ați grăbit să răspundeți la această întrebare și răspunsurile dumneavoastră menționează consecvența, claritatea și posibilitatea de a păstra elemente similare din alte zone ale sitului, sunteți pe calea cea bună a abordării celei mai mari provocări a proiectării interfeței. Alte exemple concrete sunt păstrarea paletei de culori și a fonturilor alese, păstrarea orientării vizitatorului și menținerea consecvenței navigării prin pagini.

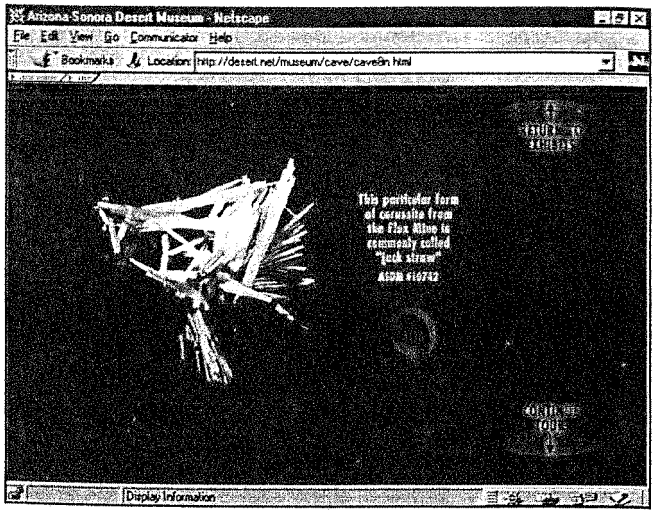


FIGURA 31.8. *Secvențele multimedia pot ocupa porțiuni mari din spațiul disponibil pe o pagină.*

Proiectarea interfeței – limbajul standard HTML

Acum, după ce ați parcurs conceptele teoretice și de planificare, este timpul să începeți aplicarea ideilor dumneavoastră creative și a rezultatelor cercetărilor în practica proiectării sitului Web. Conform metodei KISS, ideea este să pornim de la simplu și să construim pe această bază.

Se poate construi o interfață atractivă și practică folosind limbajul HTML standard. Nu vorbesc despre tabele fanteziste, cadre sau grafică complexă și tehnici multimedia; este vorba de proiectarea accesibilă, directă, pe care chiar și un începător în proiectarea Web o poate face bine. Aceasta presupune, în plus față de înțelegerea interfețelor, și cunoașterea conținutului unei pagini standard.

În acest subcapitol, vă voi prezenta trei elemente importante, necesare pentru o pagină de conținut și, pe parcurs, vă voi arăta câteva trucuri și vă voi da câteva sfaturi pentru ca pagina dumneavoastră, deși simplă, să arate într-adevăr bine. Apoi, voi integra aceste informații cu conceptul deja știut al paginii de deschidere, voi atașa un formular simplu la produsul final, și iată! Veți avea o interfață Web proprie. De fapt, putem spune că stăpâniți concepția unui sit Web.

Elementele paginii standard

Pentru a aborda problemele navigării, orientării, claritații și furnizării de informații funcționale în paginile dumneavoastră, începeți prin a verifica existența următoarelor elemente:

- **Antetul:** Un preambul declară numele sau scopul paginii. Anteturile pot fi create foarte simplu, cu text, dar este preferabil să folosiți grafica care încorporează unele elemente de stil din concepția paginii. Preambulurile permit claritate și orientare, spunând cu exactitate vizitatorului unde anume este în sit.
- **Conținutul mesajului (textul):** Textul unei pagini satisface necesitatea pentru conținut, interes și experiență în sit. Pentru a fi clar, textul trebuie să fie bine scris, concis și să folosească un ton adecvat subiectului.

- **Navigarea:** Indiferent dacă folosiți butoane, o hartă imagine, navigarea prin text sau o combinație a acestora, este important să aveți siguranța că vizitatorul sitului dumneavoastră poate ajunge oriunde cu trei clicuri de mouse.
- **Identificarea sitului:** Fiecare pagină trebuie să posede elemente care fac cunoscut vizitatorului numele sitului, întărind orientarea și claritatea. Un loc potrivit pentru a face acest lucru este în eticheta `<TITLE>`, astfel încât numele sitului și pagina respectivă să fie identificate în partea de sus a browserului. O altă metodă bună – care este în același timp și practică – pentru a identifica situl este furnizarea informațiilor privind drepturile de autor în partea de jos a paginii. Aceste informații identifică situl și furnizează o dovadă legală a drepturilor dumneavoastră de autor.
- **Contactul:** Posibilitatea de a lua legătura cu proprietarul sitului sau cu persoanele de contact corespunzătoare este foarte importantă. Prin crearea unei legături *mailto* pe fiecare din paginile dumneavoastră, întăriți imediat proiectul interfeței cu utilizatorul din punct de vedere funcțional. Pentru informații suplimentare asupra legăturilor *mailto*, vezi Capitolul 12, „Să înțelegem locatorii uniformi de resurse".

Asta-i totul? veți întreba. Răspunsul, dacă ne referim la o pagina obișnuită, este un simplu „da!"

Studiu de caz: Cornucopia

Fiind date aceste componente, priviți Figura 31.9, unde am creat o pagină cu conținut fictiv, care are la bază toate conceptele de interfață prezentate mai devreme în acest capitol ca și elementele descrise în subcapitolul anterior. Folosind limbajul HTML simplu și elemente gratuite de clip art din Caboodles (`http://www.caboodles.com/`), pagina ilustrează ce poate fi făcut chiar de către un începător care are unele idei creatoare și cunoștințe de bază corespunzătoare.

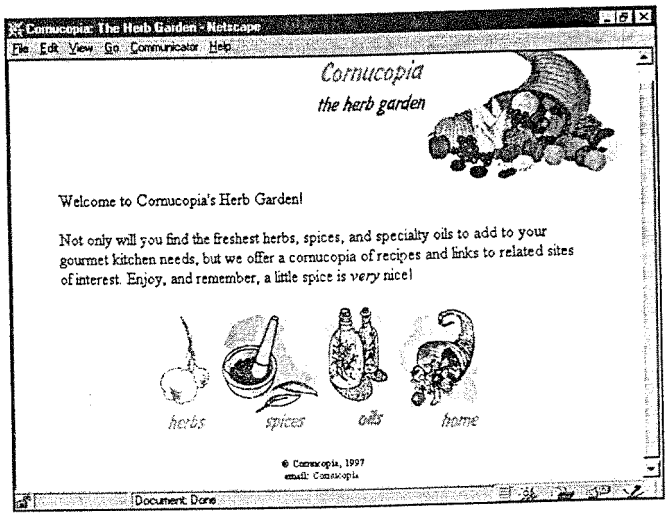


FIGURA 31.9. *Grădina de plante oferită de Cornucopia (cornului abundenței).*

Scopul acestui sit este în primul rând vânzarea specialităților alimentare oferite de firmă. Publicul este destul de larg. Gătitul este o pasiune care nu ține cont de naționalitate, sex sau vârstă. Din cauza bazei demografice largi, cea mai bună variantă de realizare a sitului este folosirea unei concepții și a unei interfețe standard, simple.

Voi examina componentele interfeței sitului și apoi le vom discuta cu referire la aplicație. O privire asupra suportului HTML vă va ajuta în primul rând să vedeți cum arată la bază interfața HTML standard cu utilizatorul.

- **Metafora:** Exemplul acestei pagini prezintă destul de clar folosirea metaforei. Deoarece situl reprezintă un magazin de vânzare cu amănuntul a specialităților alimentare, am pornit de la conceptul cornului abundenței, sugerând abundența, culoarea și gustul picant, precum și relația firească cu fiecare din plantele, condimentele și produsele reprezentate vizual în anteturi și navigare.
- **Claritatea:** Observați că am ales o metaforă simplă. Nimic nu este abstract; fiecare element ales pentru grafică poate fi identificat global. Metafora cornului abundenței garantează faptul că vizitatorul este clarificat perfect asupra a ceea ce încerc să exprim în acest sit.
- **Consecvența:** Am fost consecvent prin crearea unor imagini logice, folosind un singur fel de font pentru anteturi și navigare și păstrând gama de culori. Nu puteți vedea culorile din acest exemplu, dar voi intra în amănunte despre paleta de culori în Capitolul 33, „Folosirea culorilor și a graficii în Web". O altă modalitate de a păstra consecvența acestui exemplu de sit poate fi văzută puțin mai târziu, când voi adăuga o pagină de deschidere și un formular de reacții. Veți vedea cum voi repeta elementele pentru a obține coeziunea cerută de interfețele eficiente cu utilizatorul.
- **Orientarea și navigarea:** Antetul identifică nu numai numele sitului, dar și subiectul particular al paginii, „The Herb Garden" (grădina plantelor). Puteți obține același rezultat final folosind eticheta `<TITLE>`. Fiecare element de navigare are nume și reprezentare vizuală clare. În final, am introdus în partea de jos a paginii o notă privind drepturile de autor, pentru identificarea sitului.

Cornucopia: structura HTML

În afară de faptul că, în mod intenționat, situl este simplu datorită publicului căruia i se adresează, el constituie și un exemplu a ceea ce se poate face cu foarte puțină practică în HTML. Este puțin probabil ca situl să câștige un premiu pentru designul de avangardă, dar exemplul nu își propune să arate acest lucru. De fapt, un design modern sau mai agresiv ar putea fi un impediment pentru publicul larg și pentru experiența accesibilă pe care o doresc vizitatorilor.

Iată codul pentru exemplul de pagină. Acordați atenție modului de folosire a titlului, a variabilelor *alt* pentru imagini și a elementelor pentru drepturile de autor și *mailto*.

Listingul 31.1. Exemplu de structură HTML: Cornucopia.

```
<html>
<head>
<title>Cornucopia: The Herb Garden</title>
</head>
<body bgcolor="#FFFFFF" text="#000000" link="#CC0000" vlink="#66FF00"
alink="#FFFFFF">

<img src="corn-hed.gif" width=315 height=140 border=0 align="right"
alt="cornucopia header">
<br clear="all">

<blockquote>

Welcome to Cornucopia's Herb Garden!
<p>

Not only will you find the freshest herbs, spices, and specialty oils to add
```



```
to your gourmet kitchen needs, but we offer a cornucopia of recipes and links
to related sites of interest. Enjoy, and remember, a little spice is
<i>very</i> nice!
<p>

<blockquote>

<div align="center">

<nobr>

<a href="herbs.htm"><img src="herb.gif" width=57 height=120 border=0
➥ alt="herbs"></a>

<a href="spice.htm"><img src="spice.gif" width=103 height=120 border=0
➥ alt="spices"></a>

<a href="oils.htm"><img src="oils.gif" width=67 height=125 border=0
➥ alt="specialty oils"></a>

<a href="home.htm"><img src="home.gif" width=91 height=120 border=0
➥ alt="return home"></a>

</nobr>
<p>

<font size=1>
&copy Cornucopia, 1997<br>
email: <a href="mailto:corny@cornucopia.biz">Cornucopia</a>
</font>

</div>

</body>

</html>
```

Adăugarea unei pagini de deschidere și a formularului de reacții

Pentru a întregi exemplul și a vă crea o bună imagine a ceea ce poate face o interfață simplă, am creat o pagină de deschidere pentru Cornucopia și un formular de reacții pentru contactul cu publicul.

Pagina de deschidere, prezentată în Figura 31.10, folosește un corn simplu al abundenței, care reprezintă o introducere vizuală plină de culoare. Dedesubt, am pus un subtitlu care indică scopul sitului. În final am inclus legături de tip text către principalele secțiuni ale sitului, inclusiv către pagina de conținut prezentată mai devreme. Am plasat aceste legături pe pagina de deschidere pentru a asigura claritatea și am optat pentru text în detrimentul legăturilor grafice, deoarece doresc să pun accentul pe tipurile de produse și informațiile disponibile în sit.

În Figura 31.11, puteți vedea formularul de reacții creat. Observați că am păstrat intact antetul folosit anterior, deși am schimbat subtitlul de identificare a paginii. Am aliniat de asemenea marginile din stânga ale câmpurilor de intrare și am păstrat marginile din dreapta la minim.

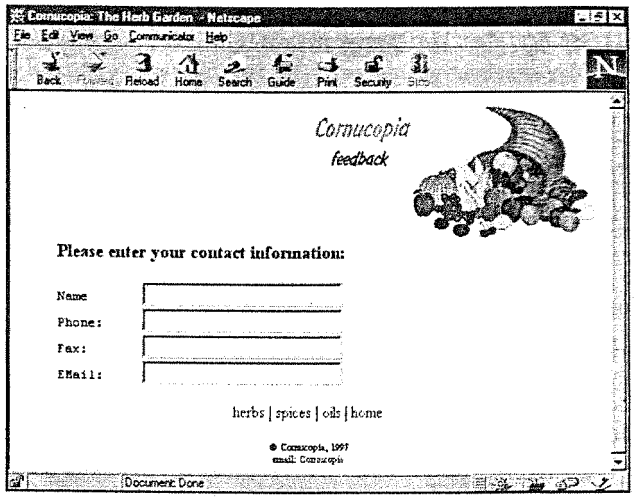
Rezultatul final este o interfață ușor de folosit, simplu de construit și care atinge scopul propus.

OBSERVAȚIE

Pentru a crea margini pe pagina în care nu utilizez tabelele, folosesc întotdeauna eticheta `<BLOCKQUOTE>`. Această metodă asigură margini în jurul textului prin furnizarea *spațiului alb*, o parte importantă a proiectării vizuale efective.

FIGURA 31.10.
Ecranul cu pagina de deschidere pentru Cornucopia.

FIGURA 31.11.
Un formular de reacții bine proiectat.



Proiectarea interfețelor sofisticate

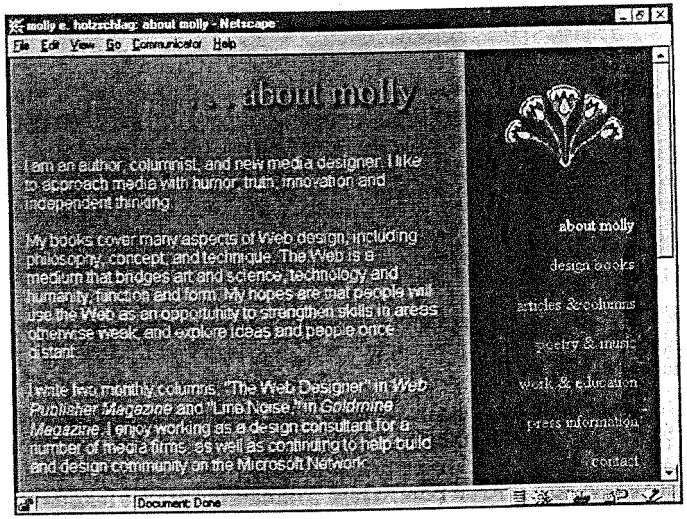
Pe măsură ce deveniți adeptul folosirii limbajului HTML, puteți trece la procedee avansate, ca de exemplu tabelele și cadrele, pentru a avea o interfață eficientă cu situl. După cum ați văzut anterior în exemplele Microsoft, această abordare poate fi foarte eficientă pentru publicul potrivit.

În acest subcapitol vă voi prezenta un exemplu cu o interfață bazată pe tabele și voi parcurge aceeași analiză ca în cazul interfeței standard. Motivul pentru care repet această etapă este de a vă lăsa să comparați și să vedeți diferența dintre două interfețe diferite care au succes datorită faptului că sunt construite din *aceleași* componente conceptuale.

Studiu de caz: Molly.Com

Voi folosi una din ipostazele propriului meu sit Web, care poate fi găsit prin mutarea browserului dumneavoastră la `http://www.molly.com/`. Deoarece publicul meu este mai bine precizat decât cel din exemplul precedent, știu că pot utiliza tabelele precum și un aspect mai abstract, ca în Figura 31.12, care este o pagină de conținut din sit. Acest lucru arată cât de importante sunt întrebările preliminare privind publicul și scopul, indiferent de tipul sitului la care lucrați.

FIGURA 31.12.
O pagină de conținut din molly.com.

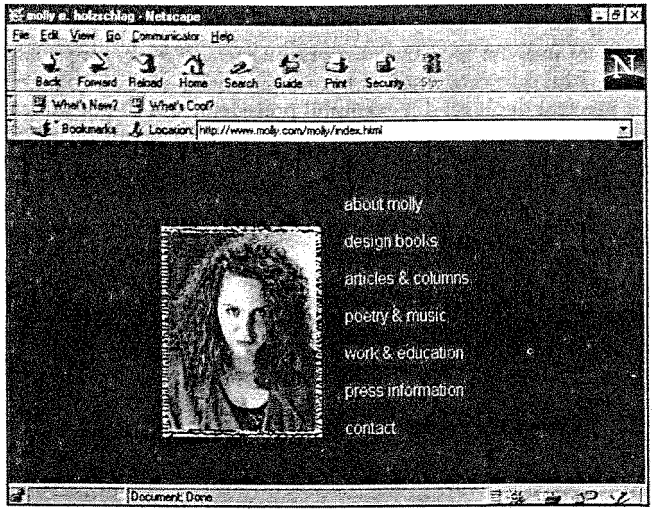


- **Metafora:** Observați că sunt numai două elemente grafice pe pagină. Unul este antetul paginii, iar celălalt un desen de mână reprezentând floarea de lotus. Floarea, care apare în același loc pe toate paginile de conținut, este un element abstract, dar consecvent al sitului. Metafora exprimă atingerea feminină și este în legătură cu munca mea în domeniul esteticii (designului) și al exprimării creative (scris și muzică).
- **Claritatea:** Deși am folosit în mod subtil metafora, fiecare pagină este clar definită, iar conținutul este direct. Această claritate permite oamenilor accesul la informația din sit fără a-și pune întrebări cum este, de exemplu, cea despre tema sitului.
- **Consecvența:** Toate paginile interioare au același aspect, iar pagina de deschidere, prezentată în Figura 31.13, conține elemente de culoare și de navigație care se regăsesc pe fiecare pagină a sitului. Aceste elemente, împreună cu tipurile de font pentru anteturi și pentru text, selectate cu grijă, determină consecvența sitului.
- **Orientarea și navigarea:** În acest sit nu folosesc navigarea grafică. Fiecare pagină este accesibilă prin lista de legături din marginea din dreapta. Anteturile marchează clar conținutul fiecărei pagini și orice pagină principală din sit este accesibilă printr-un singur clic.

Molly.com: structura html

Lucrul cu tabelele poate fi, pentru un creator de pagini Web, primul pas spre proiectarea avansată. Combinarea unor aspecte ale interfețelor, precum consecvența și textura vizuală, cu stilurile de tabele reprezintă un excelent punct de plecare. Această combinație vă va oferi, în cele din urmă, cele mai eficiente metode disponibile pentru crearea sitului dumneavoastră.

FIGURA 31.13.
Pagina de deschidere a sitului `http://www.molly.com`.



Codul acestui sit, prezentat în Listingul 31.2, este ceva mai complex decât cel din exemplul Cornucopia, dar un programator mediu sau avansat poate obține relativ ușor rezultate similare.

Indiferent de nivelul dumneavoastră de pricepere, acordați atenție mărită etichetei `<TITLE>` și atributului *alt* din imagini, care adaugă claritate și le identifică în cazul browserelor nongrafice, și atributelor pentru tipurile de font. Observați că am folosit fontul Times pentru navigare și Arial pentru text. Și anteturile grafice folosesc Times, ceea ce creează o textură vizuală frumoasă, dar oferă și consecvență.

SUGESTIE

Dacă fonturile apar prea largi sau prea înguste, le puteți adapta simplu, selectând opțiunea Font din meniul View de pe bara cu instrumente. Ambele browsere, Netscape și Internet Explorer, vă permit să stabiliți stilul sau mărimea fontului în secțiunile în care setați preferințele utilizatorului; totuși, este recomandabil să păstrați setările prestabilite;

Listingul 31.2. Exemplu de cod din situl molly.com.

```
<!-- molly e. holzschlag: molly@molly.com -->

<html>
<head>
<title>molly e. holzschlag: about molly</title>
</head>
<body bgcolor="#000000" text="#FFFFCC" link="#996666" vlink="#CCCC99"
background="images/mol-bak.gif">

<table border=0 width=600 cellpadding=5 cellspacing=0>
<tr>

<td valign=top width=400>

<img src="images/ab-hed.jpg" alt="about molly header" width=300 height=50
➥ align=right>
<br clear="all">
```



```
<pre>

</pre>

<font face=arial,helvetica>.

I am an author, columnist, and new media designer. I like to approach media
with humor, truth, innovation and independent thinking. <p>

My books cover many aspects of Web design, including philosophy, concept, and
technique. The Web is a medium that bridges art and science, technology and
humanity, function and form. My hopes are that people will use the Web as an
opportunity to strengthen skills in areas otherwise weak, and explore ideas
and people once distant.
<p>

I write two monthly columns, "The Web Designer" in <i>Web Publisher
Magazine</i> and "Line Noise," in <i>Goldmine Magazine</i>. I enjoy working as
a design consultant for a number of media firms, as well as continuing to help
build and design community on the Microsoft Network.
<p>

Other writing interests include music features, profiles, and reviews. I also
write poetry, prose, and exploratory fiction.
<p>

Music is not only a writing interest--I'm a singer, songwriter, and guitar
player with one solo album credit "Mysteries Involving Circles and Rings"
(Kept in the Dark Records, 1991). For the past several years I have been a
member of the duo Courage Sisters with Patty Sundberg.
<p>

As time goes on I realize I particularly enjoy teaching, and I am currently
developing an online masters-level course in Web design at
The New School for Social Research in New York.
<p>

I have a B.A. in writing, and an M.A. in media studies from The New School.
I am about to pursue a Ph.D. in the humanities. My studies focus on the
consciousness of cyberspace and the use of nonlinear environments as tools
for human evolution.
<p>

Travel, cats, elephants, esoteric studies, philosophy, healing arts, and the
relationship between art and science are among my personal interests.
</font>

</td>

<td valign=top align=right width=200>

<img src="images/lotusm.gif" alt="Lotus Blossom" width=125 height=125>
<p>

<a href="molly.htm">about molly</a>
<p>

<a href="books.htm">design books</a>
<p>

<a href="write.htm">articles & columns</a>
<p>
```

(continuare)

Listingul 31.2. – continuare

```
<a href="poems.htm">poetry & music</a>
<p>
<a href="work.htm">work & education</a>
<p>
<a href="press.htm">press information</a>
<p>
<a href="contact.htm">contact</a>
<p>
<a href="index.html">home page</a>
</td>
</tr>
</table>
</body>
</html>.
```

SUGESTIE

În şirurile care definesc imagini folosiţi întotdeauna valorile `height` (înălţime) şi `width` (lăţime). Acestea reprezintă un instrument de bază pentru capacitatea browserului de a furniza informaţii în mod logic.

Rezumat

Conceptele şi exemplele din acest capitol ar trebui să vă fixeze bine pe calea proiectării interfeţelor eficiente folosind limbajul HTML. Următorul capitol vă va prezenta conceptele şi metodele care vor perfecţiona deprinderile dumneavoastră în proiectarea interfeţelor.

Amintiţi-vă că înţelegerea părţilor unui întreg este importantă pentru înţelegerea întregului. Această abordare este demonstrată clar în cazul în care studiaţi concepţia interfeţei cu utilizatorul. Puteţi întări componentele interfeţei prin tratarea atentă a unor conceptelor precum metafora, claritatea şi orientarea. Analizând publicul, scopul şi tipul experienţei pe care o doriţi vizitatorilor, obţineţi o imagine generală, care vă conduce la obţinerea siturilor Web unitare, mai uşor de folosit şi, în cele din urmă, mai profesionale.

Capitolul 32
Navigarea şi machetarea

de Molly E. Holzschlag

În acest capitol:

- **Proiectarea pentru ecranul calculatorului** 480
- **Neliniaritatea şi interactivitatea** 483
- **Mediile de navigare** 484
- **Opţiuni de navigare** 487
- **Folosirea marginilor din stânga şi din dreapta pentru navigare** 489
- **Navigarea folosind cadrele pe marginile din dreapta şi din stânga ecranului** 492
- **Navigarea multiplă la lucru** 492

Navigarea şi machetarea sunt la fel de critice în proiectarea unui sit ca şi interfaţa acestuia, reprezentând de fapt o extensie a proiectării interfeţei cu utilizatorul. Acestea constituie calea prin care oamenii obţin şi cunosc conţinutul pe care-l oferiţi şi, dacă un vizitator Web nu poate face acest lucru, este ca şi cum nici nu aţi avea un sit Web! Navigarea şi machetarea, împreună cu interfaţa, ar trebui luate în considerare în etapele iniţiale ale proiectării sitului dumneavoastră. Prin combinaţia reuşită a acestora se creează premiza pentru obţinerea celor mai eficiente situri, iar proiectanţii lor devin cei mai căutaţi în domeniu.

De fapt, proiectanţii care au făcut descoperirile de pionierat în domeniul utilizării potenţialului neliniar şi interactiv al reţelei Web, ca bază pentru procedeele de navigare, au atras în mare parte atenţia pentru acest efort. Similar, proiectanţii care înţeleg limitările ariei vizuale a unui sit şi creează proiecte care folosesc la maxim potenţialul ecranului calculatorului se plasează în frunte.

De ce este navigarea aşa de importantă? veţi întreba. Nu este aceasta doar o modalitate de a oferi opţiuni pentru capitolele sitului şi care permite oamenilor să ajungă în aceste zone prin intermediul legăturilor? Ei bine, acest lucru este adevărat, dar există mai multe complicaţii – sau poate *ocazii* ar fi cel mai potrivit cuvânt – care au un rol în crearea navigării pe Web.

Primul aspect este destul de clar: multe situri Web sunt enorme! Ele conţin mii, dacă nu zeci sau chiar sute de mii de pagini de date. Dacă nu sunteţi convins de acest lucru, priviţi, de exemplu, situl Microsoft de la adresa `http://www.microsoft.com/`.

Alt aspect, pe care-l veţi vedea disecat şi examinat în acest capitol, tratează publicaţiile online. Ziarele şi revistele sunt preocupate de găsirea unei modalităţi de lucru cu ştirile secundare, care conţin adesea informaţii relevante. Pe măsură ce se adună pagini de conţinut, creşte nevoia de administrare a datelor, atât în culise cât şi la vedere.

În final, mediul neliniar Web creează o problemă specială pentru cei obişnuiţi cu proiectarea paginală cu pagină. Având în vedere posibilităţile oferite de hipermedia, legături şi forme multiple de navigare, opţiunile pentru proiectant şi utilizatorul final se diversifică. Un sit Web nu este o carte, deci pentru a obţine informaţia nu treceţi la pagina următoare sau daţi o pagină înapoi. Ei bine, *puteţi* prezenta şi în acest fel informaţia, dar nu este

obligatoriu! Trebuie să stăpâniți această neliniaritate și să oferiți opțiuni și lucruri interesante vizitatorilor sitului dumneavoastră.

OBSERVAȚIE

Un alt exemplu de mediu neliniar este discul CD-ROM interactiv, de exemplu o enciclopedie, unde anumite cuvinte și imagini au hiperlegături cu alte informații, asemănător celor din situl Web.

Pe lângă aspectele legate de navigare, proiectanții siturilor trebuie să ia în considerare și modul de machetare a paginilor Web. Luați o carte, un ziar, o broșură sau un fluturaș pentru reclamă și veți vedea cum cuvintele și imaginile sunt etalate pe hârtie în diferite moduri. Privind reclamele comerciale TV, veți obseva diferențele dintre modalitățile de prezentare a informației vizuale pe ecran.

Concepția de aranjare a proiectelor Web nu este diferită de cea a oricărei forme familiare de informare. Sunt diferite numai restricțiile specifice pentru ecranele calculatoarelor și pentru mediul tehnologic în general.

În capitolul anterior, „Realizarea unei interfețe eficiente", am lansat ideea că interfața este modalitatea prin care dumneavoastră comunicați și interacționați cu informația de pe ecran. Am menționat numai aspectele navigării, care sunt importante nu numai pentru studiul concepției interfeței, dar și pentru machetele paginilor Web.

Motivul pentru care navigarea influențează machetarea este nu numai acela că designerii trebuie să învețe să creeze interfețe logice și machete familiare, însă interesante, dar și că elementele de navigație trebuie incluse în pagină. O reclamă din ziar sau TV nu solicită deloc acest lucru, fapt care introduce o nouă provocare pentru proiectanții din mediul Web.

Acest capitol vă prezintă noțiuni de bază importante privind machetarea și se concentrează pe mediile și metodele de navigare. Țineți minte noțiunile învățate privind studiul interfețelor când parcurgeți și studiați exemplele de față și veți începe să înțelegeți cum se îmbină toate acestea, nu numai pentru a forma componentele unei pagini Web, dar și un sit cu servicii complete.

Proiectarea pentru ecranul calculatorului

După cum probabil știți, există multe platforme de calculatoare, două dintre ele, MacIntosh și PC, dominând piața comercială. În afară de aceasta, există o mulțime de plăci video și monitoare. Rezultatul final este că niciodată nu puteți fi sigur cine vă privește siturile și cum le vede.

Designerii profesioniști de aplicații Web folosesc o varietate de metode pentru a rezolva problema platformelor multiple. Una dintre aceasta este să îți imaginezi unde se plasează persoana cea mai obișnuită și să încerci să proiectezi folosindu-l pe acesta ca model.

Din acest punct de vedere, este în general adevărat că cea mai mare parte dintre oameni vizualizează Webul la o rezoluție de 640x480. Aceasta înseamnă că spațiul vizibil total este lat de 640 de pixeli și lung de 480 de pixeli. Însă, această suprafață nu corespunde zonei de proiectare pe care trebuie machetată munca dumneavoastră, pentru că browserul ia ceva din acest spațiu! Ca exemplu, deschideți într-un browser oricare pagină cu bara completă de instrumente afișată, și veți vedea că se pierde spațiu în părțile de sus, laterale și de jos ale ecranului. Acest lucru face de fapt lățimea puțin mai mică de 640 și reduce lungimea paginii.

SUGESTIE

Designerii trebuie să seteze rezoluția ecranului la 640x480 și să folosească acest mediu pentru crearea și testarea paginilor. Este bine să vedeți paginile și la o rezoluție mai mare, dar faceți asta numai după ce sunteți sigur că ați lucrat bine în limitele 640x480.

Proiectarea pe orizontală și verticală

Pe lângă restricțiile impuse de browsere asupra spațiului, mai există și problema proiectării pe verticală și orizontală. Orice este mai mare de 600 de pixeli fie că va forța apariția unei bare orizontale de defilare (după cum se prezintă în Figura 32.1), fie se va transforma în ceva încurcat, total diferit de cea ce v-ați propus să creați. Așa că regula de bază pentru machetarea pe orizontală este să stabiliți lățimea sitului la 600 de pixeli. Sunt unii designeri care se aventurează dincolo de această limită și proiectează „la orizont", dar trebuie să fiți un foarte bun designer pentru a face aceasta și, de aceea, pentru majoritatea creatorilor de pagini Web nu pot recomanda acest lucru.

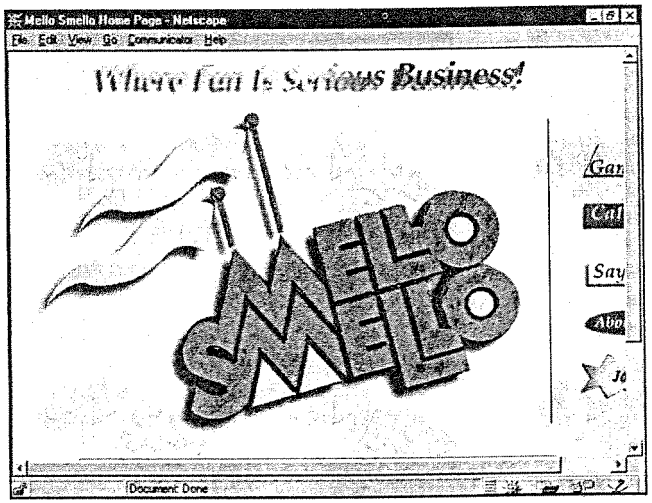


FIGURA 32.1.
O bară de defilare orizontală care nu ar trebui să fie acolo – lucru pe care proiectanții buni caută să-l evite.

OBSERVAȚIE

Uneori designerii pot ieși cu proiectul din spațiul pe orizontală, dar ei trebuie să aibă un motiv întemeiat și o pricepere deosebită pentru a reuși. Un exemplu interesant de proiectare de acest tip poate fi găsit la adresa `http://www.circleoffriends.org/`. (vezi Figura 32.2)

Într-adevăr, o lățime de 600 de pixeli pare destul de mică, mai ales dacă obișnuiți să vedeți paginile Web la rezoluții mai mari. Totuși, până când se va modifica rezoluția obișnuită, veți dori să ajungeți la cât mai mulți oameni. Singura excepție obișnuită este cazul în care sunteți sigur că toți cei care vizitează situl dumneavoastră folosesc o rezoluție mai mare. Ei bine, în acest caz puteți proiecta la acele dimensiuni. Un exemplu poate fi mediul intranet, unde utilizatorii au aceleași configurații hardware și software.

Proiectarea pe verticală este ceva mai flexibilă, deoarece utilizatorii din Web folosesc în mod natural defilarea în jos pentru a vedea mai multe informații. Există numai două probleme în proiectarea pe verticală.

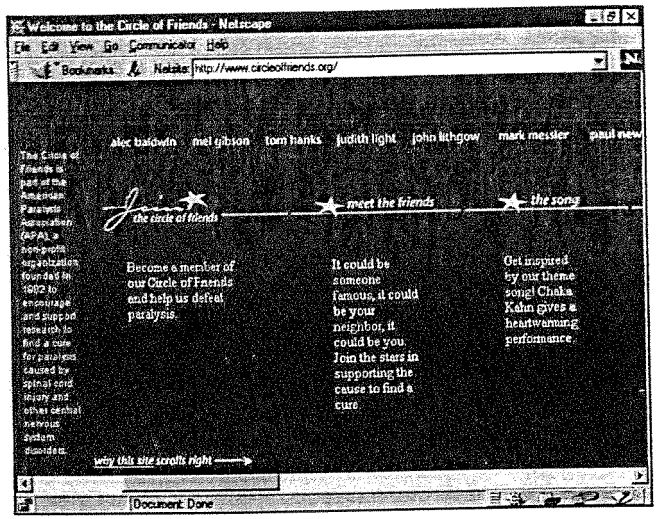


FIGURA 32.2. *Bara de defilare pe orizontală are sens deoarece acest sit este proiectat pentru a demonstra că mulți pacienți cu leziuni ale nervului spinal văd în mod diferit lumea.*

Prima este aceea că în oricare pagină de deschidere doriți să aranjați informațiile importante pe un singur ecran, înainte de defilare. Cu alte cuvinte, o imagine foarte lungă care solicită vizitatorului defilarea pentru a o vedea poate fi neplăcută și urâtă pe pagina de bază (vezi Figura 32.3). În cazul paginilor interioare, unde textul și grafica se amestecă într-o manieră fluidă, mai puțin statică, dispuneți de mai multă flexibilitate.

FIGURA 32.3. *Această imagine de deschidere este trunchiată din cauza restricțiilor pe verticală.*

Cealaltă problemă a proiectării pe verticală este numărul de ecrane cu care ar trebui permisă defilarea. Sugestia mea este că mai mult de trei face pagina prea lungă; în acest caz trebuie să luați în considerare mutarea informației pe altă pagină. Dacă aveți o pagină lungă, o cale agreabilă de a face viața utilizatorilor dumneavoastră mai ușoară este să oferiți legături înapoi către partea de sus a paginii (vezi Figura 32.4). Acest lucru este

adevărat în mod special când se ține cont de navigare, accesul convenabil la opțiunile de navigare facând benefică experiența vizitatorilor sitului.

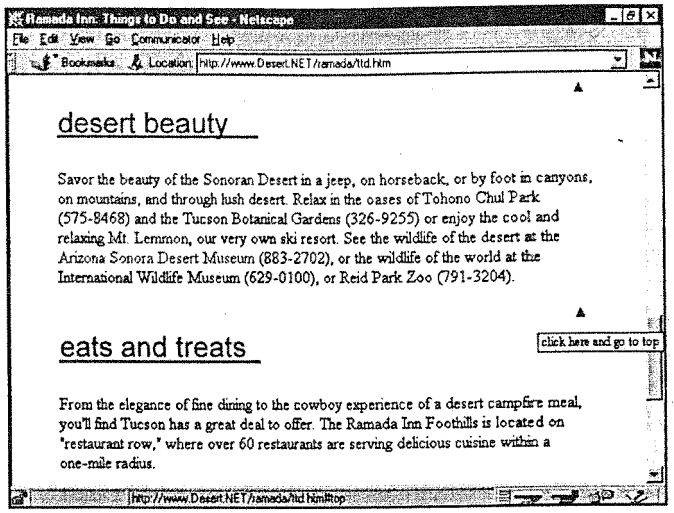


FIGURA 32.4. *Legăturile înapoi către partea de sus a paginii sunt agreabile pentru vizitatorii din Web.*

Deși considerațiile privind machetarea sunt importante, vă puteți descurca cu noțiunile de bază prezentate. Pentru moment sunt destul de limitate, dar aceasta nu înseamnă că nu veți învăța despre noțiuni mai avansate privind machetarea, cum sunt, de exemplu, cele prezentate în Capitolul 34, „Folosirea spațiului, a formei și a tipului caracterelor". Este important să vă formați o idee asupra domeniului în care lucrați și să înțelegeți mai întâi cum se integrează navigarea în imaginea de ansamblu.

Neliniaritatea și interactivitatea

Unul dintre aspectele care au cea mai mare influență în Web este că acesta e un mediu neliniar, interactiv. Acest lucru înseamnă că, spre deosebire de ziare, radio și televiziune, Webul nu are la bază un sistem de livrare pagină înainte-pagină înapoi. Un ziar în limba engleză sau multe alte limbi e citit de la stânga la dreapta. Radioul și televiziunea furnizează informațiile într-un pachet pe care vi-l servesc fără să vă implice prea mult. Singurele dumneavoastră opțiuni curente, interactive pentru radio și TV sunt să le porniți, să modificați volumul, să schimbați canalul și să le închideți.

Pe măsură ce tehnologiile Web și cele legate de acesta converg, se va înregistra probabil o schimbare în planul dominant, neliniar. Pentru moment, în locul unui număr redus de opțiuni, rezultat din concepția liniară, variante multiple de navigare pot controla această structură neliniară, oferind o experiență personală cu interactivitate mare vizitatorilor Web.

Din punct de vedere conceptual, liniaritatea e ca drumul dumneavoastră de dimineață la serviciu. De regulă folosiți același drum deoarece știți că ajungeți repede, traficul e mai puțin aglomerat în comparație cu cel de pe alt drum, iar ritmul traficului vă este cunoscut. Neliniaritatea ar presupune folosirea unui drum diferit în fiecare zi – apucând pe un drum necunoscut doar de dragul de a o face, poate chiar mergând pentru o vreme într-o direcție diferită, deoarece peisajul e atractiv.

Acest exemplu vă ajută să vedeți avantajele și dezavantajele liniarității și neliniarității. Nu din întâmplare multe societăți s-au bazat pe liniaritate. Drumul direct, cel mai eficient a fost necesar pentru supraviețuire și prosperitate. El a fost avantajos în multe feluri, dar o dată cu rigiditatea lui s-au pierdut oportunitățile pentru perspective și posibilități noi, care pot fi găsite prin explorarea drumului lateral.

Din punct de vedere filozofic, acest fenomen cultural conduce la posibilitatea ca o combinație de metode liniare și neliniare să constituie cea mai bună metodă de abordare a problemei. Dacă aveți ceva timp în plus, ați putea să mergeți astăzi la serviciu pe un drum diferit. Cine știe ce va ieși din acest lucru?

Deoarece suntem familiarizați cu liniaritatea, tangentele neliniare ale sitului Web – hiperlegăturile care ne scot dintr-un sit și ne aruncă în mijlocul altuia și opțiunile multiple de navigare, care fac din fiecare parcurgere a autostrăzii informaționale o experiență diferită – reprezintă o provocare pentru proiectantul de pagini Web. Acesta trebuie să pună frâu naturii tangențiale a mediului Web pentru ca vizitatorul să poată percepe scopul și logica sitului. În același timp, el știe că prin realizarea unui sit liniar se pierd posibilități ale mediului Web, locuri și experiențe neobișnuite pe care acesta le poate oferi.

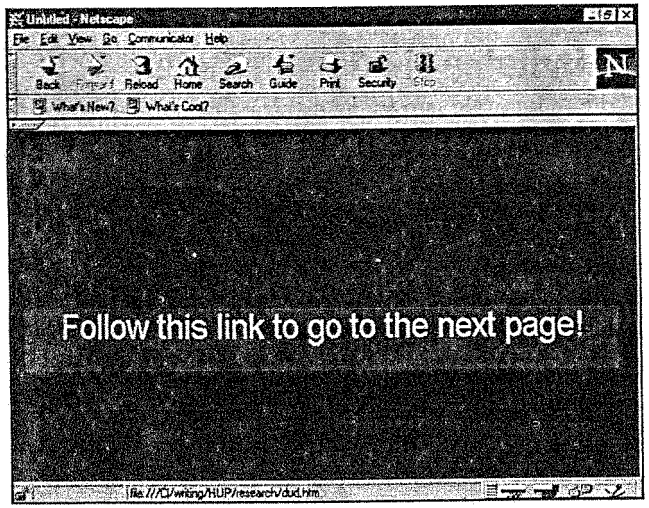
Mediile de navigare

Deși sunt multe posibilități de navigare, mediile grafice și metodele HTML necesare pentru realizarea navigării sunt limitate. Următoarele elemente sunt considerate ca făcând parte din mediile standard de navigare:

- **Legături tip text:** Acestea sunt obișnuitele ancore, având la bază limbajul HTML, care permit selectarea unei porțiuni de text și crearea unei hiperlegături între aceasta și alt document Web (vezi Figura 32.5). Iată un exemplu de legătură tip text:

```
<a href="nextpage.html>Follow this link to go to the next page!</a>
```

FIGURA 32.5.
Iată un exemplu de legătură prin text.



- **Marcaje de listă și imagini separate:** imaginile separate pot avea hiperlegături prin introducerea lor într-o ancoră HTML. Proiectantul poate creea în acest scop marcaje de listă (Figura 32.6) sau o imagine decorativă (Figura 32.7). Codul corespunzător graficii destinate

unei hiperlegături este prezentat mai jos. Observați că, în acest caz, codul corespunzător imaginii ia locul textului prezentat în exemplul precedent, cu legătură standard prin text.

```
<a href="nextpage.html><img src="images/nextpage.gif"></a>
```

FIGURA 32.6.
Marcajele grafice de listă atrag atenția privind legăturile fierbinți.

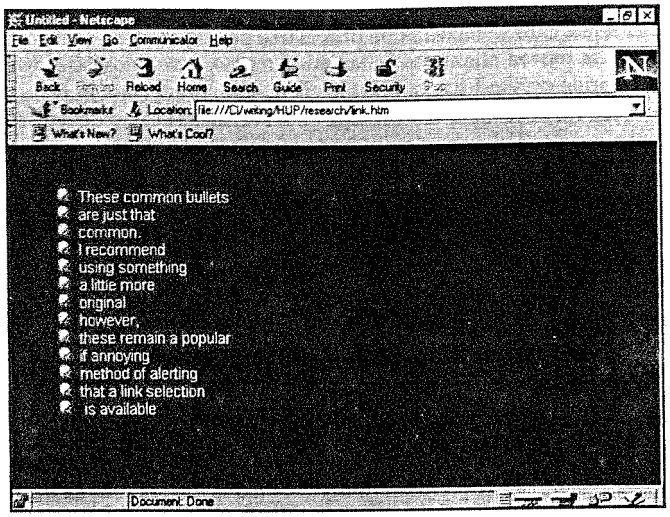
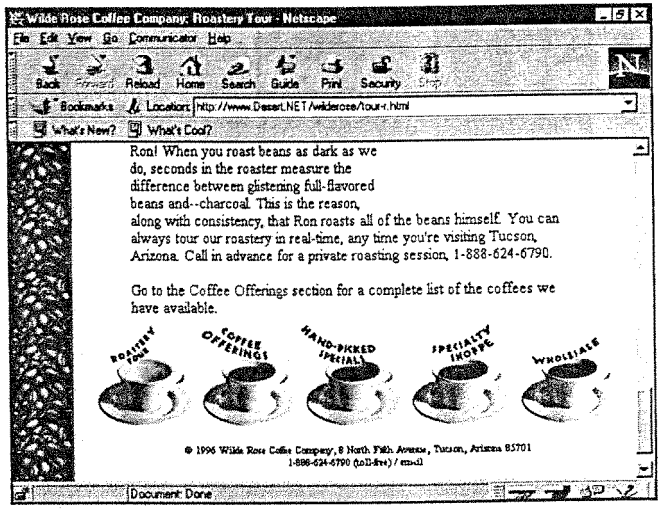


FIGURA 32.7.
Butoane decorative – o abordare mult mai sofisticată.



OBSERVAȚIE

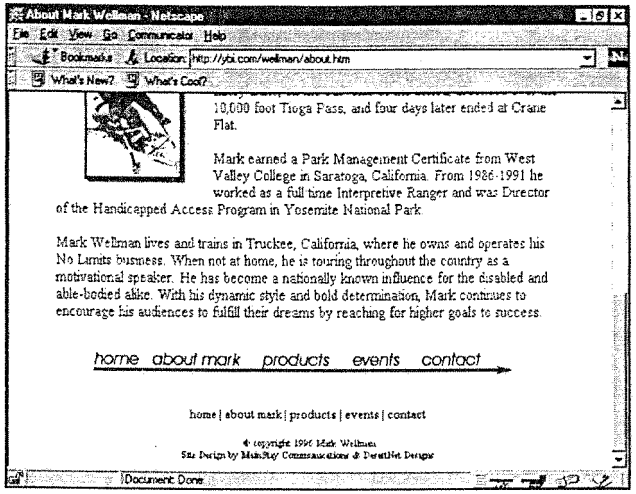
Crearea unui identități a zonelor prin schimbarea graficii destinate navigării vă ajută să diferențiați o pagină de alta. Aceast procedeu obișnuit de proiectare joacă un rol în relația dintre navigare și conceptul de orientare întâlnit la proiectarea interfeței. Observați că una dintre cestile de cafea din Figura 32.7 e goală, simbolizând identitatea paginii.

■ **Hărți imagine:** O hartă imagine este constituită în esență dintr-o singură imagine și din cod pentru ancora HTML corespunzător anumitor coordonate ale proiectului hărții. Prin utilizarea acestei metode trebuie încărcată numai o singură imagine – dar care are posibilități multiple de navigare.

Cele mai cunoscute tipuri de hărți imagine sunt două: hărțile de partea serverului (server-side maps), care se bazează pe procesarea de către server a datelor privind coordonatele și hărțile de partea clientului (client-side maps), care folosesc browserul pentru a prelucra informațiile privind harta. Hărțile client-side, prezentate în Figura 32.8, sunt mai rapide, deoarece nu necesită o conversație cu serverul, așa că au devenit soluția standard. Iată un exemplu de cod client-side din harta imagine precedentă:

```
<MAP name="marcmap">
<AREA shape="rect" coords= "0,0,53,30" href="index.htm">
<AREA shape="rect" coords= "53,0,162,30" href="about.htm">
<AREA shape="rect" coords= "162,0,255,30" href="products.htm">
<AREA shape="rect" coords= "255,0,327,30" href="events.htm">
<AREA shape="rect" coords= "327,0,410,30" href="contact.htm">
<AREA shape="rect" coords= "0,0,440,46" nohref>
</MAP>
```

FIGURA 32.8.
O hartă imagine client-side – textul, săgeata și umbrele sunt toate părți ale unei imagini.



SUGESTIE

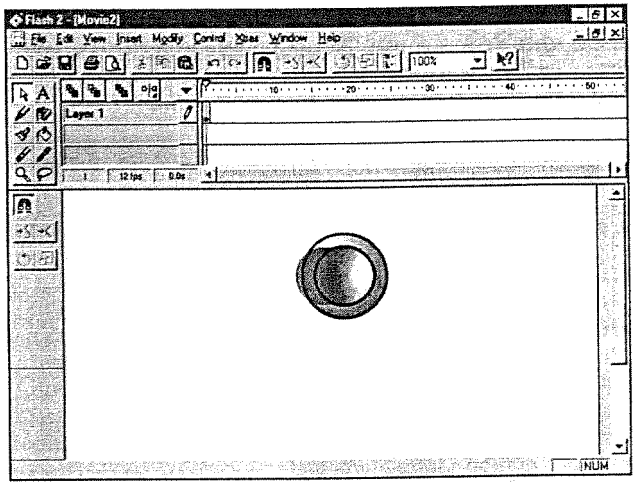
Pentru detalii în legătură cu hărțile server-side, consultați administratorul de sistem sau furnizorul de servicii Internet (ISP) în legătură cu componentele necesare pentru a crea astfel de hărți. Amănunte privind hărțile client-side pot fi găsite la adresa `http>//www.spyglass.com/techspec/tutorial/img_maps.html`.

■ **Medii avansate:** Alte posibilități de hiperlegătură presupun folosirea unei diversități de medii cu aceeași funcție ca a legăturilor prin text sau prin butoane, dar asemănătoare codului din hărțile imagine. Procedeele corespunzătoare mediilor avansate cuprind

Macromedia Flash și Shockwave, în cazul cărora butoanele și hărțile pot fi animate și create cu legătura codată direct în mediu (Figura 32.9) și codul de bază inclus în limbajul HTML, astfel:

```
<object classid="clsid:D27CDB6E-AE6D-11CF-96B8-444553540000"
➥codebase="fsplash.cab" align="baseline" border="0" width="70" height="35">
<param name="Movie" value="button.swf">
<param name="Loop" value="False">
<param name="Play" value="True">
<param name="BGColor" value="ffffff">
<param name="Quality" value="high">
<param name="Scale" value="Showall">
</object>
```

FIGURA 32.9.
Acest buton animat a fost creat cu Macromedia Flash.



Alte medii avansate pot fi create folosind aplicații ale programării avansate, cum sunt Java, VBScript, ActiveX și VRLM (Virtual Reality Modeling Language- Limbajul de modelare al realității virtuale). Aceste metode avansate necesită un nivel de pregătire mult mai înalt față de cel necesar în cazul metodelor directe, bazate pe text, grafică sau hartă. Totuși, rezultatele pot răsplăti pe deplin efortul, deoarece acestea adaugă noi niveluri de funcționalitate la metodele convenționale de navigare întâlnite în cazul majorității siturilor Web.

SUGESTIE

Căutați informații suplimentare privind posibilitățile de navigare avansate? Pentru metodele Flash și Shockwave vizitați `http://www.macromedia.com/`. Situl de origine pentru limbajului Java al firmei Sun Microsystems, `http://www.javasoft.com/`, abundă de informații privind acest limbaj. Resursele pentru VBScript și ActiveX pot fi găsite pe Microsoft Sitebuilder Network, la adresa `http://www.microsoft.com/sitebuilder/`.

Opțiuni de navigare

O dată cu selectarea mediilor, un designer trebuie să aleagă și locul în pagină al opțiunilor de navigare. Pe măsură ce machetarea prin intermediul limbajului HTML devine mai flexibilă, același lucru se întâmplă și cu variantele aflate la dispoziția designerului.

De obicei, designerul alege un traseu principal de navigare, cum ar fi poziționarea sus, jos, pe marginea din stânga și pe cea din dreapta. Apoi poate adăuga alte opțiuni, care au ca rezultat mai multe posibilități pentru vizitatorul sitului Web – și, în consecință, mai multă interactivitate.

Folosirea părților de jos și de sus ale ecranului pentru navigare

Probabil că cel mai obișnuit sistem de navigare este cel cu opțiunile plasate în partea de jos a ecranului. Acestea pot fi create cu oricare din mediile de navigare standard, cu o metodă de machetare tradițională sau cu tabele și cadre. Aceste posibilități vă oferă flexibilitate maximă în momentul în care vă hotărâți să abordați un anumit proiect Web.

De ce se folosește partea de sus a ecranului pentru navigare? Motivul este creearea unei interfețe navigabile *fără* a întrerupe spațiul prețios al conținutului primului ecran din pagina Web. Însă navigarea din partea de sus a ecranului trebuie tratată cu atenție, deoarece zona de sus constituie unul dintre primele locuri pe care cade în mod natural privirea. Dacă aveți acolo elemente care distrag atenția vizitatorului de la scopul principal al sitului, riscați să pierdeți interesul vizitatorului.

OBSERVAȚIE

Cei care folosesc zilnic Internetul pot citi lucruri interesante despre el vizitând Start Reading. Pe lângă faptul că articolele acoperă mai multe teme vizionare și răsunătoare, pagina este și un exemplu elocvent pentru folosirea părții de sus a ecranului, astfel ca navigarea să nu interacționeze cu alte elemente ale paginii. Îl puteți găsi la `http://home.microsoft.com/reading/voices.asp`.

Această măsură de prevedere nu înseamnă că nu puteți folosi partea superioară a ecranului pentru navigare. Totuși, trebuie să procedați cu grijă, pentru a nu interacționa cu alte informații importante care ar putea fi poziționate la începutul paginii. Asigurați-vă că navigarea este ușor de folosit și este bine integrată în arhitectura sitului. Dacă integrați elemente și aveți alte articole imediate care să distragă privirea de la navigare, veți obține un echilibru mai bun între elementele de navigare și cele referitoare la conținut.

SUGESTIE

Cum pot face ca marginile aliniate sus să apară în partea de sus a browserului? Puteți face acest lucru prin controlul marginii superioare cu eticheta `<BODY>`, dar aceasta nu poate fi citită decât de browserul Microsoft, Internet Explorer (IE). Paginile de stiluri sunt compatibile atât cu IE 3.0, cât și cu Netscape 4.0 și le puteți folosi pentru a stabili marginea superioară la poziția 0.

Aspectul „cum" al navigării aliniate în partea de sus a ecranului e la fel de direct ca și cel al variantelor pentru partea de jos a ecranului, atât în ceea ce privește machetarea, cât și mediile, cum sunt legăturile prin text, butoane grafice și hărți imagine. Puteți realiza cu cel mai mare succes acest lucru folosind cadrele sau tabelele, care vor fi prezentate în paragraful următor; fiecare metodă are un potențial remarcabil și setul normal de particularități.

Folosirea părții superioare a ecranului pentru navigare: metode

În paginile în care controlul poziționării e un element critic sunt folosite adesea tabele. Acestea permit proiectantului să fixeze locul destinat fiecărei componente a proiectului. Butoanele pot fi poziționate, ca și hărțile imagine și chiar textul.

Și cadrele sunt adeseori folosite în partea de sus a ecranului pentru navigare. Avantajul cadrelor este că generează o zonă de navigare care rămâne fixă pe durata vizitei sitului

Web. Spre deosebire de paginile de informație, care se reîncarcă, cadrul rămâne pe loc. Această facilitate permite nu numai existența unui punct constant de referință, dar și scăderea timpului de încărcare a unei pagini, deoarece accentul cade acum numai pe încărcarea informației, nu pe navigare.

SUGESTIE

Două surse bune de informații pentru navigarea cu tabele și cadre sunt siturile de bază ale browserelor. Netscape poate fi găsit la `http://home.netscape.com/`, iar Microsoft are rețeaua vastă Site Builder Network pentru designeri Web la adresa `http://www.microsoft.com/sitebuilder/`.

Când se folosesc cadrele pentru navigare, o greșeală obișnuită este crearea unui cadru mic pentru o cantitate mare de informație. Cu alte cuvinte, dacă creați un cadru cu înălțimea de numai 75 de pixeli, verificați dacă în cadrul acestor limite încape conținutul împreună cu ceva spațiu liber în jur. Altfel, veți ajunge să forțați oamenii să citească datele dintr-un spațiu mic. Conținutul înghesuit nu este recomandabil din punct de vedere al proiectării și nu este practic și convenabil pentru vizitatorii sitului dumneavoastră Web.

Folosirea părții de jos a ecranului pentru navigare: metode

Metodele standard de navigare folosesc limbajul simplu HTML, textul, grafica sau hărțile imagine pentru a realiza proiectul. Aceste metode sunt directe și uneori elegante, dar nu oferă un control la fel de eficient ca acela care folosește tabelele sau cadrele.

Când folosiți tabele pentru un sistem de navigare plasat în partea de jos, reamintiți-vă lecțiile privind proiectarea interfețelor. Cum puteți realiza în cele mai bune condiții un proiect interesant? Tabelele vă oferă un control foarte bun asupra elementelor individuale – grafică, text și navigare – astfel încât să considerați pagina ca un tot unitar. Folosirea părții de jos a ecranului pentru navigare ar trebui să se îmbine cu alte elemente ale paginii pentru a crea o adevărată interfață.

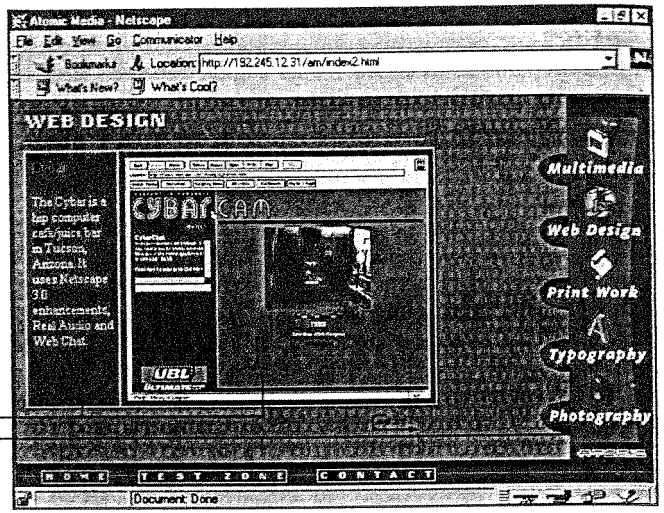
Pentru metoda bazată pe cadre, avantajele sunt similare cu cele corespunzătoare folosirii părții de sus a ecranului pentru navigare. În primul rând, navigația e fixă; ea poate fi pe orice pagină doriți, fără a necesita reîncărcare și rămâne consecventă chiar dacă treceți de la conținutul unei pagini la al alteia. Aveți de asemenea flexibilitate în crearea cadrelor cu sau fără margini (vezi Figura 32.10). Și, ca și în cazul sistemului de navigare cu cadre situat în partea de sus a ecranului, un cadru cu margini vă oferă varianta de redimensionare pentru controlul deplin al ecranului.

Folosirea marginilor din stânga și din dreapta pentru navigare

O metodă obișnuită de proiectare a navigării este cea care folosește metode standard, bazate pe tabele sau bazate pe cadre pentru controlul unui sistem de navigare plasat pe o margine.

Asemănător variantelor prin care se folosesc părțile de jos sau de sus ale ecranului, exemplele de navigare folosind marginea din dreapta sau stânga permit o mare flexibilitate în ceea ce privește mediile. Navigarea folosind cadrele păstrează fixă porțiunea respectivă din pagină, în timp ce secțiunea țintă încarcă informația. Controlul maxim al proiectului se realizează folosind tabelele.

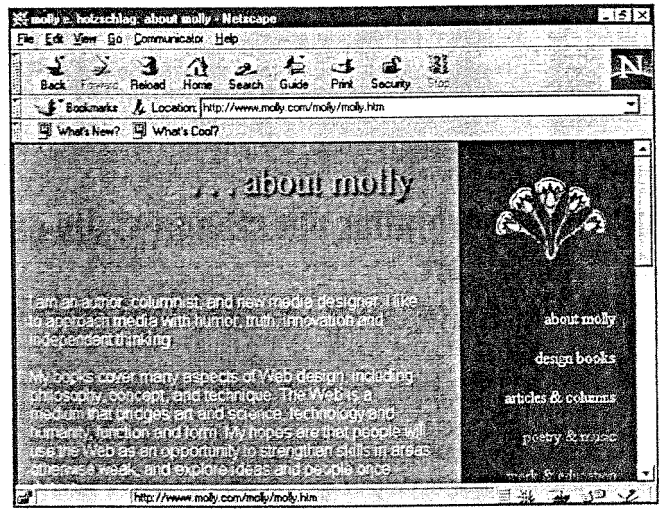
FIGURA 32.10. *Această pagină folosește tabele și cadre fără margini pentru controlul unei interfețe compacte și elegante.*



Metode care folosesc marginea din stânga sau dreapta pentru navigare

E important să subliniem importanța uneia dintre cele mai simple convenții Web: fundalul cu o margine de tip tabel. Ați întâlnit des această metodă – o bandă evidențiată prin culoare și concepție, situată de-a lungul marginii din stânga sau din dreapta și plasată lângă un câmp texturat sau cu culoare compactă sau un câmp de textură care încorporează conținutul textului (vezi Figura 32.11).

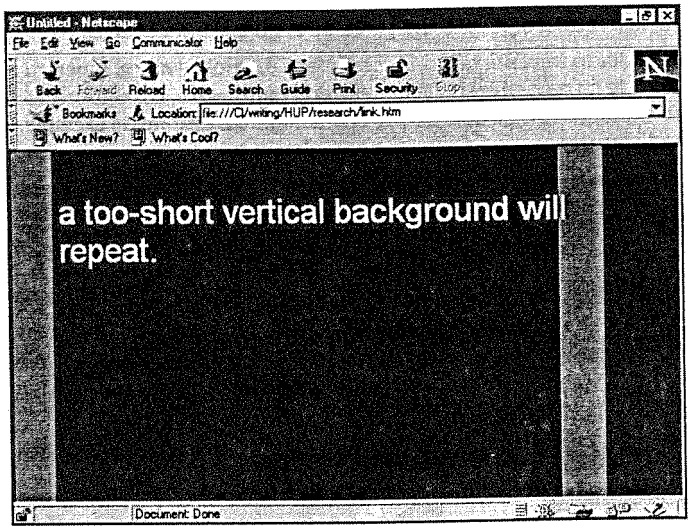
FIGURA 32.11. *Un fundal cu margine de tip tabel.*



După cum am menționat în capitolul precedent, referitor la machetare, proiectantul trebuie să acorde atenție restricțiilor care privesc proiectarea pentru platforme multiple. Există mai multe aspecte privitoare la fundaluri.

Vă recomand să construiți toate fundalurile la o dimensiune de 1024 de pixeli sau mai mare! Mă contrazic singură, veți întreba? Nu chiar. Luați în considerare faptul că cei mai mulți utilizatori folosesc lățimea de de 640 de pixeli, dar *următoarea* valoare uzuală de lățime pentru Web este de 800 de pixeli. Valori mai mari ale rezoluției, de 1280 sau peste, nu sunt obișnuite. Totuși, dacă nu anticipez posibilitatea unor astfel de rezoluții înalte când este vorba despre fundaluri, mă voi confrunta, în final, cu problema repetiției, după cum se vede în Figura 32.12.

FIGURA 32.12. *Un fundal prea îngust se va repeta.*



OBSERVAȚIE

Doriți să cunoașteți amănunte suplimentare privind fundalurile? Pagina cu întrebări frecvente privind fundalul, întreținută de Mark Koenen, e disponibilă la adresa `http://www.sci.kun.nl/thalia/guide/color/faq.html`.

Apoi, plasați marginea într-un câmp pe care-l puteți controla. În cazul marginii din stânga, acest lucru înseamnă să începă la extremitatea din stânga și să se termine într-un loc care să lase suficient spațiu pentru conținut, în condițiile folosirii rezoluției mici. Deci întreaga margine nu va fi, de regulă, mai lată de 200 de pixeli, cu aproximativ 400 de pixeli rămași în câmpul din dreapta pentru text și conținut.

Similar, fundalul cu marginea în dreapta trebuie să rezerve câmpului pentru text și conținut lățimea necesară (aproximativ 400-450 de pixeli). Marginea din dreapta începe din acel punct și se continuă cu valoarea rămasă până la totalul de 1024. Procedând astfel, ați rezolvat o parte a problemei care forțează apariția barelor de defilare orizontale și evitați repetarea fundalului.

Navigarea folosind cadrele pe marginile din dreapta și din stânga ecranului

Ca și în cazul folosirii părții de jos sau de sus a ecranului pentru navigare, avantajul navigării cu ajutorul cadrelor este acela că rămâne o parte fixă. Dacă alegeți marginea din stânga sau din dreapta, cu sau fără chenar, acest lucru depinde numai numai de cerințele și gusturile dumneavoastră privind proiectarea și machetarea.

Toate variantele de navigare se confruntă cu aceleași probleme privind spațiul, iar luarea în considerare a spațiului alocat proiectului este imperativă. Cît spațiu avem la dispoziție? În afară de faptul că trebuie să țineți cont de limitările privind pixelii, trebuie să testați comportarea produsului dumneavoastră la folosirea cu diferite tipuri de browsere, pentru a vedea dacă există mici diferențe de la un exemplu la altul.

Aveți acum informațiile necesare selectării oricărei margini pentru navigare și creării unor interfețe cu navigare și machetare corespunzătoare. Următorul pas este acela de a examina modul de folosire a acestor procedee într-o diversitate de exemple, pentru a demonstra în mod clar puterea machetării și navigării în mediul Web.

OBSERVAȚIE

Navigarea nu reprezintă singura utilizare a cadrelor și tabelelor. Puteți folosi cunoștințele după cum doriți – nu trebuie să limitați aplicațiile la ceea ce ați învățat aici. Un exemplu obișnuit de folosire a cadrelor și tabelelor în afara navigării îl reprezintă reclamele.

Navigarea multiplă la lucru

Vă voi prezenta acum o metodă care poate fi aplicată cu succes pentru a controla structurile și interactivitatea neliniară și a profita de avantajele lor. S-ar putea să fie un exemplu cunoscut pentru aceia dintre dumneavoastră care mi-ați urmărit lucrările, pentru că-l folosesc adeseori pentru demonstrațiile privind proiectarea neliniară, bazată pe opțiuni.

Cel mai bun exemplu din Web pentru a demonstra metodele multiple de navigare rămâne Tucson Weekly. De fapt, publicația și-a extins cu timpul interactivitatea. Poate intrarea timpurie a acestei publicații în mediul Web, combinată cu inteligența plină de inovație a proiectantului acesteia, Wil Gerken, i-a conferit acesteia poziția superioară în comparație cu alte publicații și situri Web care încearcă, dar au dificultăți în realizarea neliniarității.

OBSERVAȚIE

Vizitați publicația Tucson Weekly la adresa `http://www.tucsonweekly.com/tw/`.

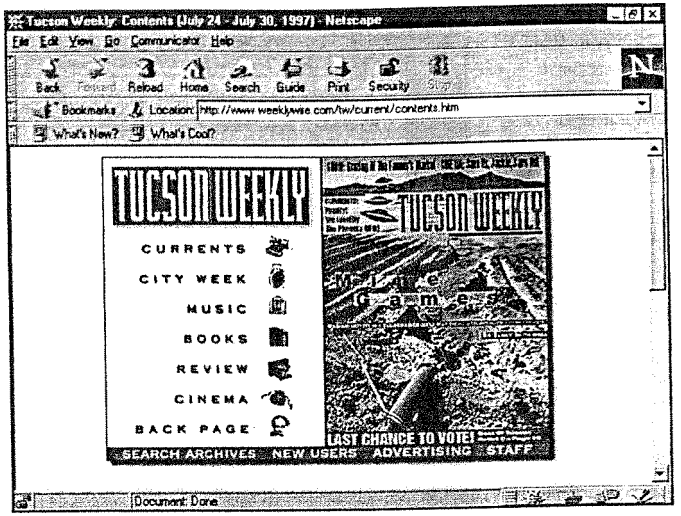
Scopul acestui paragraf este să vă ofere o privire în profunzime, prezentându-vă argumentele pentru care navigarea este fantastică în a oferi vizitatorului Web un sit eficient care captează, dar nu controlează mediul tangențial aflat în spate.

Modul principal prin care Tucson Weekly a realizat conceptul „captare fără control" este prin furnizarea metodelor multiple de navigare în sit și prin oferirea căilor de ajungere la alte situri înrudite, dar care nu aparțin publicației.

Iată o privire generală asupra structurii și navigării în publicație, precum și asupra rezultatelor obținute:

■ **Pagină de deschidere (splash) pentru intrare:** Pagina de deschidere reprezintă modalitatea principală de acces la ziar și la secțiunile acestuia. Se modifică în fiecare săptămână pentru a corespunde cu coperta respectivă. (Vezi Figura 32.13)

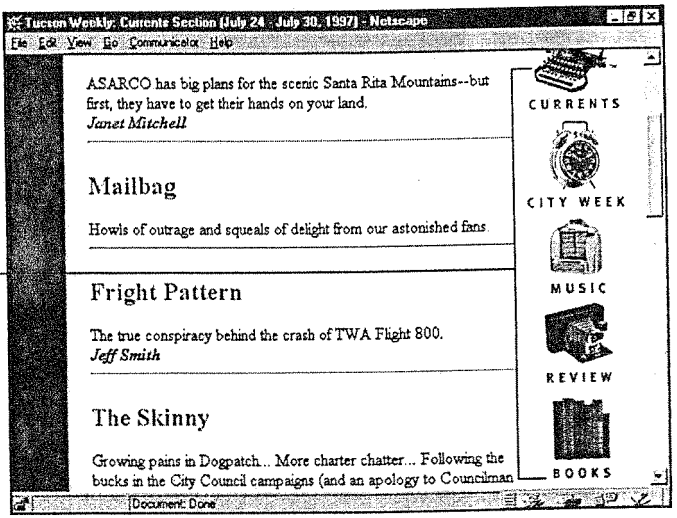
FIGURA 32.13. *Pagina de deschidere a publicației Tucson Weekly.*



■ **Navigarea folosind un tabel pe marginea din dreapta a ecranului:** Secțiunile importante ale ziarului sunt prezentate într-o hartă imagine plasată pe marginea din dreapta, așa că puteți trece la altă secțiune a ziarului, indiferent dacă sunteți într-o pagină de secțiune sau conținut (vezi Figura 32.14).

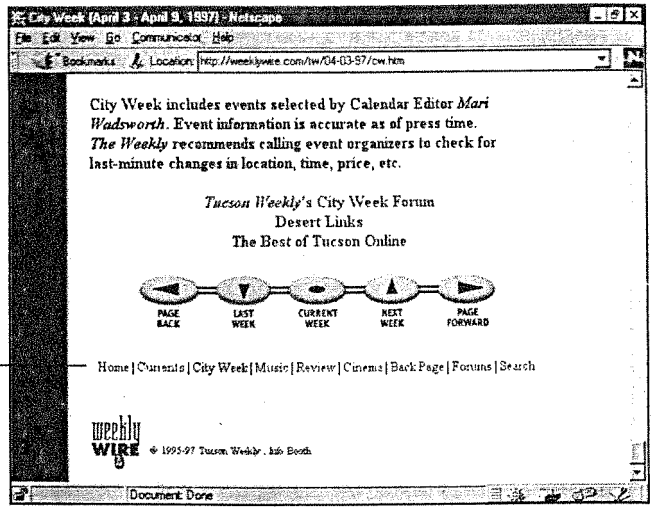
FIGURA 32.14. *Harta imagine din partea dreaptă a ecranului, folosită pentru accesarea altor zone ale sitului.*

O hartă imagine pe marginea din dreapta



■ **Opțiunea de folosire a textului în partea de jos a marginii de navigare din dreapta:** Secțiunile principale pot fi accesate prin metoda folosirii textului în partea de jos a paginii (Vezi Figura 32.15).

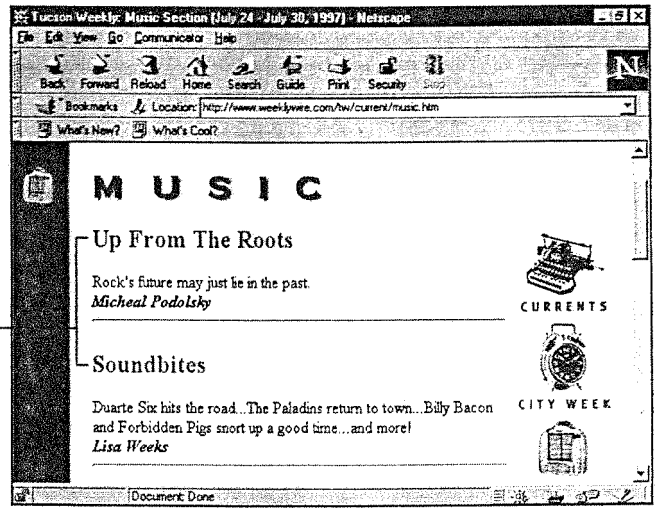
FIGURA 32.15. *Legături prin text în partea de jos a ecranului, folosite pentru a accesa secțiunile principale ale ziarului.*



Legătură prin text în partea de jos a ecranului

■ **Legături prin text la paginile interne:** Secțiunile sunt împărțite în legături la articole și forumuri online asociate din cadrul sitului Tucson Weekly (vezi Figura 32.16).

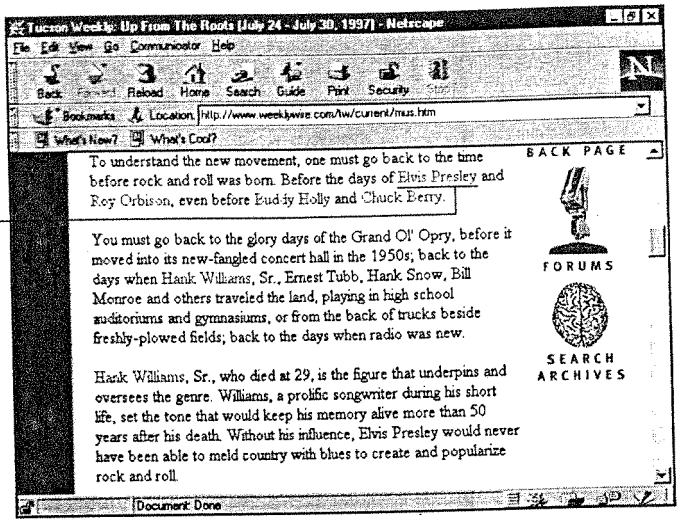
FIGURA 32.16. *O pagină de secțiune cu legături prin hipertext la articole.*



Legături de tip antet

■ **Legături din interiorul paginilor:** Legăturile prin text din interiorul paginilor ies afară în Internet pentru a oferi informații referitoare la un anumit subiect dezbătut în articol, după cum se vede în Figura 32.17.

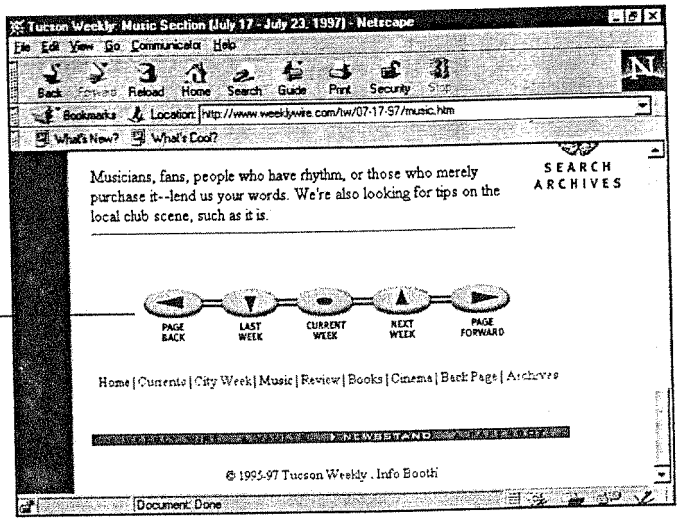
FIGURA 32.17. *Un articol cu legături hipertext în Internet.*



Legătură de tip text

■ **Meniul principal din paginile interioare:** Acest meniu reprezintă cununa de lauri a navigării prin ziar. Oferă o opțiune liniară de forma pagină înainte- pagină înapoi. Apoi, adaugă tangente neliniare către ediția săptămânii viitoare (dacă sunteți în secțiunea de arhivă a ziarului), ca și un „salt" către ediția din săptămâna respectivă. Nu contează unde vă aflați în timp și spațiu – data sau locul publicației – puteți accesa orice alt punct al ziarului cu câteva clicuri de mouse (vezi Figura 32.18).

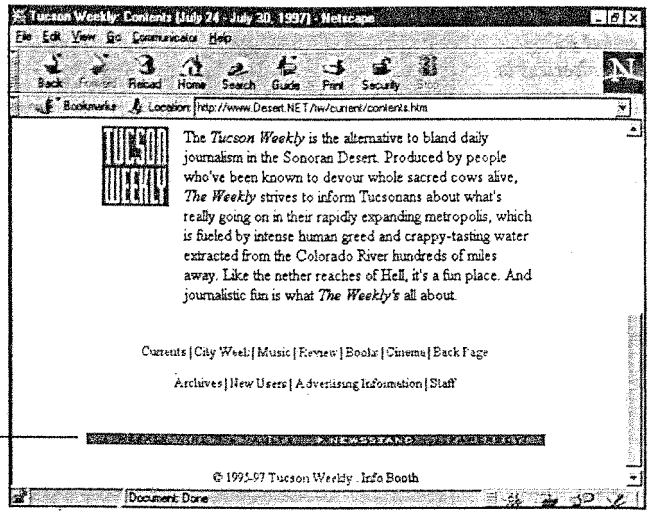
FIGURA 32.18. *Exemplul strălucitor privind navigarea neliniară din Tucson Weekly.*



Meniul principal dintr-o pagină interioară

■ **Meniul Weekly Wire:** Un menu grafic care vă oferă alăturate incomparabilele produsele ale companiei de publicații online, inclusiv pagina de bază The Weekly Wire și proiectele surori (vezi Figura 32.19).

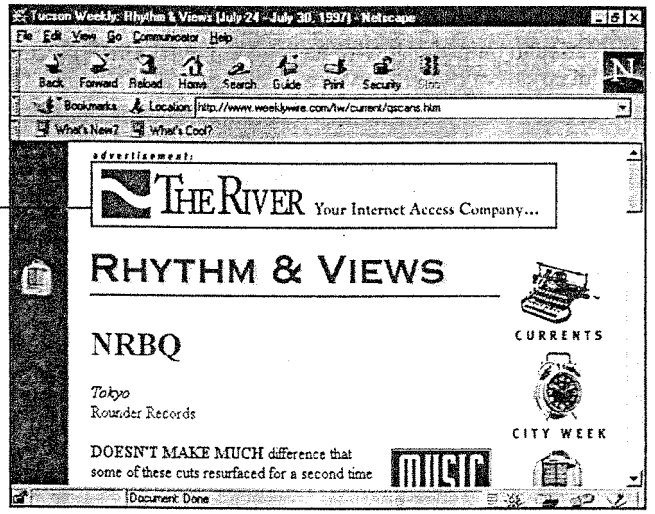
FIGURA 32.19. *Meniul Weekly Wire.*



Grafică pentru navigare

■ **Reclamele:** În partea de sus a paginilor ziarului Tucson Weekly, este rezervat spațiu pentru firmele locale sau naționale, specializate în reclame (vezi Figura 32.20).

FIGURA 32.20. *Spațiul din Tucson Weekly destinat publicității.*



O legătură destinată publicității

OBSERVAȚIE

Puteți studia alt caz privind navigarea neliniară (realizat de aceiași designeri pentru alt ziar) vizitând publicația Albuquerque Alibi, la adresa `http://www.desert.net/alibi/`.

Sunteți obosit? Eu da! Țineți minte că Tucson Weekly e elaborat săptămânal, disponibil pe situl Web miercuri seara, *înaintea* ediției tipărite, și – iată magia – necesită numai câteva ore pe săptămână pentru a fi pus în pagină. Cum reușesc, rămâne un secret profesional, dar

acest lucru trebuie să vă îndemne la îndrăzneală și originalitate în gândire. Folosiți exemplul privind machetarea și navigația inteligentă din acest paragraf pentru a ajunge la noi performanțe în proiectare.

Rezumat

Este ușor de înțeles cum atâtea opțiuni ar putea de fapt să vă distragă de la satisfacția explorării unui sit Web. Tucson Weekly demonstrează ceea ce ați învățat în acest capitol prezentând o metodă sofisticată, multiplă de navigare. Este captat mediul neliniar, iar relațiile dintre el și procesul de gândire liniar ajută vizitatorul să trăiască interactivitatea fără să se rătăcească.

Sunt folosite elementele standard, precum meniul de navigare din partea dreaptă a ecranului, meniurile din partea de jos a ecranului, legăturile prin text și grafică, fundalul cu margine albastră – fiecare lucru dintre acestea servește ca bază vizuală pentru că dumneavoastră le înțelegeți. Ele nu vă sunt străine. Totuși, ceea ce e neobișnuit, vizionar chiar, este *modul* de combinare al lor. Singurul lucru demn de a fi amintit e că toate acestea pot fi realizate și de către dumneavoastră; este posibil chiar să descoperiți modalități noi, neobișnuite de abordare a proiectării Web.

Capitolul 33
Folosirea culorilor și a graficii în WEB

de Molly E. Holzschlag

În acest capitol:

- **Paletele de culori – stilul personal 498**
- **Noțiuni de bază privind culorile în limbajul HTML 502**
- **Folosirea culorilor neprotejate 504**
- **Culorile browserului 504**
- **Tehnologia grafică în Web 505**
- **Lucrul cu grafica 508**
- **Alte tehnici grafice 513**
- **Imaginile și limbajul HTML 513**

Îmi place să se spună despre mine că sunt plină de culoare. Înțeleg acest lucru ca un compliment, deoarece culoarea este plină de vibrație, energie și este interesantă. Și siturile Web ar trebui să fie pline de vibrație, energie și să prezinte interes, ceea ce le-ar face pline de viață și atrăgătoare pentru vizitatori.

Culoarea este un element esențial al designului. Totuși, folosirea ei cu succes în Web reprezintă o provocare pentru majoritatea designerilor. Controlul culorii în Web este limitat din cauza diferențelor dintre platformele de calculatoare, monitoarele video și browserele Web.

Dar stați liniștit! Designerii atenți pot urma anumite căi călăuzitoare – ușoare – care-i vor ajuta să aleagă stilul vizual și procedeele tehnice în mod corespunzător. Acest capitol vă prezintă atât conceptele, *cât și* procedeele necesare pentru a umple de culoare siturile dumneavoastră Web.

Paletele de culori – stilul personal

Începeți prin planificarea paletei personale de culoare a sitului dumneavoastră, pe care să o folosiți la selectarea culorilor pentru fundaluri, text, legături definite prin limbajul HTML, precum și pentru grafică. Alegerea culorilor pentru un anumit sit ar trebui să fie rezultatul cunoașterii publicului, a tematicii și a dorințelor clientului. Amintiți-vă de faptul că prin culoare se comunică emoție, așa că țineți cont de acest lucru când faceți alegerea.

O revistă de avangardă (vezi Figura 33.1) va avea o paletă personală de culori diferită față de un cotidian mai liniștit, dar totuși interesant (vezi Figura 33.2). Publicul revistei Wired preferă culorile vii (de tip neon), care sugerează senzația de acțiune, mișcare și echilibristică mai mult decât o fac culorile fundamentale. Acest aspect este potrivit pentru publicul revistei Wired, dar n-ar fi adecvat pentru o publicație tradițională. Arizona Daily Star folosește culori mai subtile, cu accentul pe negru, alb și albastru, cu tușeuri de roșu. Aceste culori dau senzația unui mediu proaspăt și curat. Vizitatorii sitului se concentrează asupra conținutului în mod natural, fără să acorde prea multă atenție concepției – și acest lucru este exact ceea ce doriți în cazul unui cotidian.

INFORMATICA

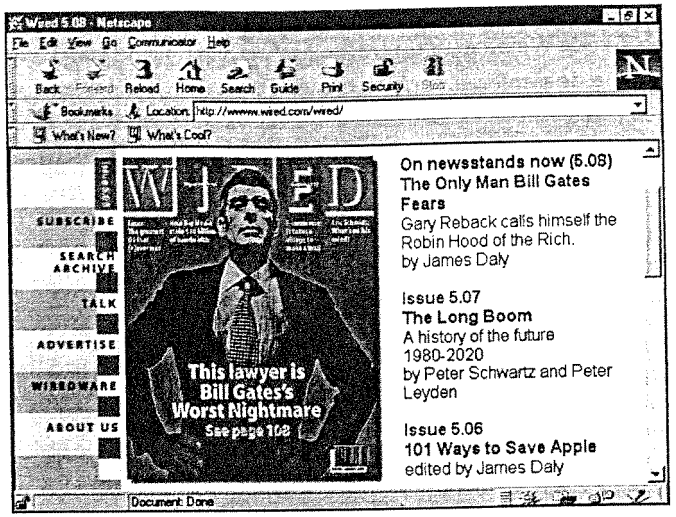


FIGURA 33.1. *Revista Wired, de la adresa* `http://www.wired.com/`, *folosește culori vii.*

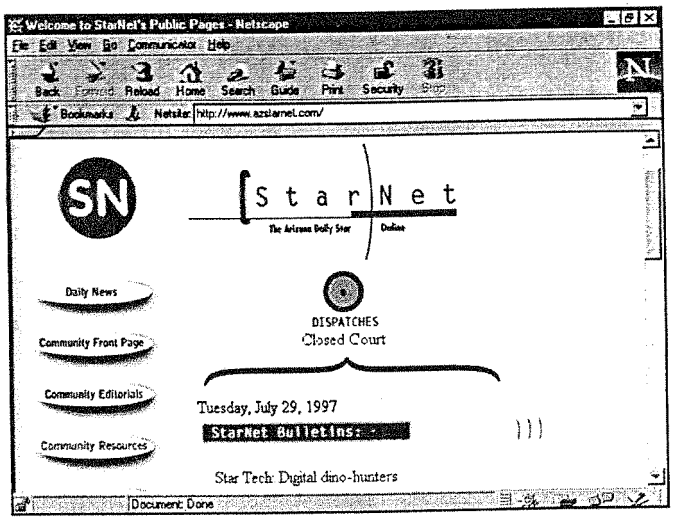


FIGURA 33.2. *O abordare mult mai calmă se observă în cotidianul Arizona Daily Star, la* `http://www.azstarnet.com/`.

Totuși, cu siguranță că nu ați dori să schimbați schemele de culori! Culorile vii pur și simplu nu-și găsesc locul în paginile cotidianelor relativ conservatoare. Dacă un anume client preferă aspectul revistei Wired, dar are un public format din oameni în vîrstă, ar fi bine să-l lămuriți ce efect vor avea aceste culori asupra vizitatorilor.

Culori protejate

Următorul pas este înțelegerea noțiunii de paletă „protejată". Aceasta este o paletă cu 21 culori care vor rămâne stabile la schimbarea browserului, platformei și a rezoluției monitorului. Utilizatorii versiunii Photoshop 4.0 sunt norocoși – acest program dispune de propria sa paletă protejată. Aceia care au alte versiuni ale produsului Photoshop pot transfera

și instala o paletă protejată, direct în program, din situl unui guru în domeniul graficii Web, Lynda Weinman, `ftp://luna.bearnet.com/ pub/lynda/`. Căutați fișierul `bclut2.aco`.

Ar fi bine, de asemenea, să obțineți fișierul `nvalue.gif`, care vă furnizează toată gama de culori protejate pentru browser, cu valorile RGB (red, green, blue) și hexazecimale corespunzătoare. Valorile RGB sunt obținute printr-o metodă care determină numeric cantitățile de roșu, verde și albastru din culoarea respectivă. Valorile hexazecimale sunt date de sistemul de numerație în baza șaisprezece, care este format din cifrele 0-9 și literele A-F. Un octet (8 biți) poate fi reprezentat folosind două caractere, ceea ce face ca orice combinație binară să fie mai ușor de înțeles. În ceea ce privește culorile Web, valorile hexazecimale au *întotdeauna* șase caractere.

Puteți afla singur valoarea hexazecimală a oricărei culori folosind calculatorul științific. Când vă aflați în programul Photoshop, treceți cu cursorul peste oricare culoare. Fereastra pop-up „info" afișează valorile individuale sub formă de numere pentru componentele roșu, verde și albastru ale culorii. (Vezi Figura 33.3.) Introduceți pur și simplu fiecare din aceste valori în calculatorul științific (deoarece folosesc Windows'95, pot utiliza calculatorul științific disponibil, după cum se vede în Figura 33.4; trebuie doar să selectați Scientific din meniul View) și selectați butonul radio numit Hex. Calculatorul vă furnizează apoi combinația alfanumerică corespunzătoare valorii culorii.

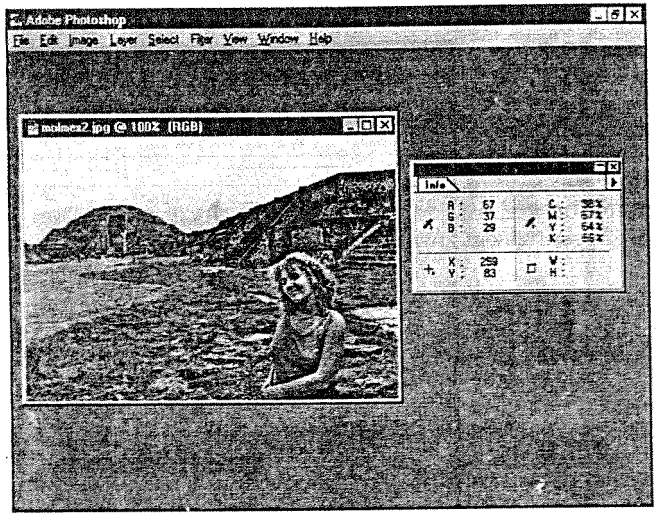


FIGURA 33.3. *Valorile RGB afișate de PhotoShop 4.0.*

Să presupunem că am o paletă care conține o nuanță medie de gri. Valoarea roșie în exemplul meu este 153; valoarea hexazecimală este 99. Acum voi parcurge același procedeu cu valorile pentru verde și albastru. Observați că valoarea 0 pentru roșu, verde sau albastru va fi scrisă în acest scop 00. Întotdeauna, la sfârșit trebuie să obțineți în total șase caractere. Rezultatul conversiei de la RGB la hexazecimal pentru nuanța medie de gri este 999999, deoarece toate valorile RGB sunt 153. Pentru alte culori se pot obține alte combinații; unele formate din aceleași cifre, altele, din perechi de cifre, cum ar fi `cc9900`.

Crearea paletei individuale

Pentru a crea o paletă individuală, selectați 3–5 culori din paleta protejată; o mostră este prezentată în Figura 33.5. Verificați dacă spectrul ales de dumneavoastră conține cel puțin o culoare deschisă și una închisă. Ideile pot fi luate din natură, obiectele înconjurătoare sau

dintr-o gamă variată de cărți de proiectare. Dacă cunoașteți dinainte care culoare predomină în situl dumneavoastră, acesta va avea un aspect unitar și elegant.

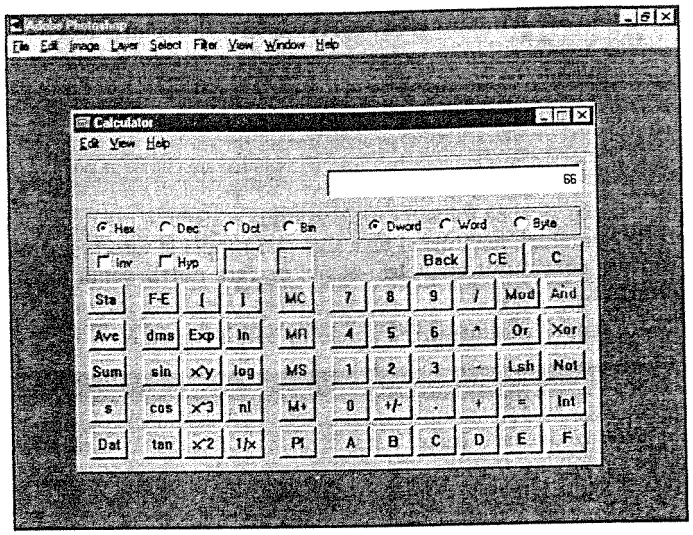


FIGURA 33.4. *Folosiți calculatorul științific din Windows 95 pentru conversia hexazecimală rapidă.*

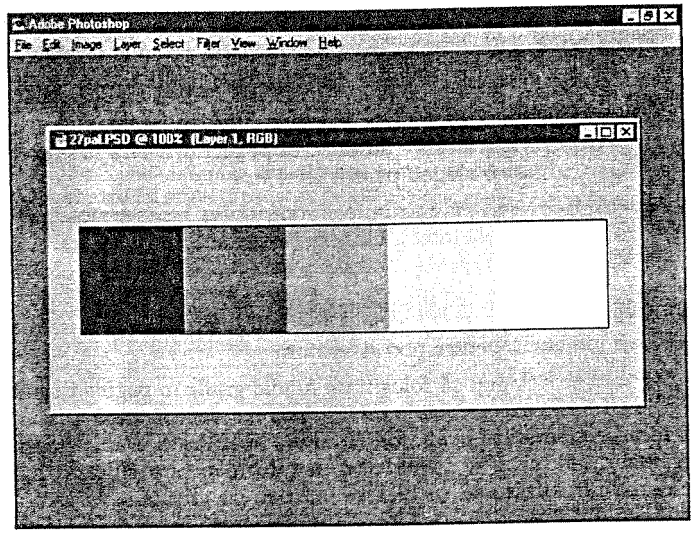


FIGURA 33.5. *În această carte am folosit pentru exemplificare o gamă de nuanțe de gri, dar ar fi bine ca dumneavoastră să vă aventurați mai departe.*

Creați palete personale în PhotoShop și salvați-le ca fișiere PhotoShop astfel încât să puteți păstra integritatea paletei. De asemenea, nu uitați să notați valorile hexazecimale ale fiecărei culori. Acest lucru vă va folosi când veți scrie codul.

Paleta dumneavoastră personală este gata de lucru!

Noțiuni de bază privind culorile în limbajul HTML

Valorile hexazecimale ale paletei pe care tocmai ați creat-o vor fi folosite în aplicațiile grafice și în codul dumneavoastră HTML. În HTML puteți avea controlul deplin al culorii, deoarece browserul interpretează informația hexazecimală repede și eficient.

SUGESTIE

La adresa `http://www.webreference.com/dlab/9704/` poate fi găsit un articol excelent, scris de Dmitry Kirsanov, care tratează aspectele tehnice ale culorii în Web.

Probabil veți dori să folosiți aceste valori la realizarea unor combinații creative de culori pentru fundaluri, legături și text. Cea mai simplă și directă aplicație pentru designeri o constituie conținutul etichetei `<BODY>`, cu care puteți defini fundalurile, textul și stilul legăturilor.

Eticheta `<BODY>` permite următoarele atribute de culoare:

OBSERVAȚIE

Limbajul standard HTML 4.0 recomandă folosirea foilor de stil în cascadă ca o metodă de obținere a culorii browserului. Totuși, deoarece majoritatea designerilor lucrează pentru un public care nu folosește browsere compatibile cu foile de stil, se folosesc vechile metode. Dacă totuși optați pentru foile de stil, este bine ca deocamdată să includeți ambele metode.

- **`bgcolor=x`**: această valoare hexazecimală indică culoarea de fond a întregii pagini.
- **`text=x`**: atributul `text` generează culoarea pentru întreg textul standard, fără legături, din pagină.
- **`link=x`**: culoarea introdusă prin acest atribut va apărea în locurile în care aveți legături prin text.
- **`vlink=x`**: o legătură vizitată va apărea în culoarea pe care o alegeți pentru acest atribut.
- **`alink=x`**: o legătură *activă* – adică aceea pe care urmează să se execute clic – va apărea în culoarea pe care o alegeți pentru acest atribut.

O etichetă `<BODY>` cu aceste atribute se prezintă după cum urmează. Veți observa că am plasat valoarea hexazecimală între ghilimele. Fiecare atribut individual începe cu semnul `#`:

```
<body bgcolor="#FFFFFF" text="#000000" link="#CCCCCC" vlink="#CCCCCC"
➥alink="#999999">
```

Figura 33.6 prezintă rezultatele. Observați diferențele gradate dintre atributele pentru culoarea fondului, pentru text și legături.

S-ar putea să decideți să folosiți un fundal grafic în pagina dumneavoastră, caz în care folosiți atributul `background="url"` în șirul precedent. Această valoare avertizează browserul să încarce elementul grafic de la adresa specificată. Dacă folosiți un fundal grafic, vă recomand călduros să includeți și o valoare pentru culoarea fundalului. În acest fel, culoarea fundalului se va încărca imediat, iar grafica se va încărca peste acea culoare. Produsul final elimină efectul neplăcut și creează un efect vizual unitar.

Iată sintaxa pentru aceeași pagină cu două coloane și grafică cu două culori poziționată în partea de sus a fundalului:

```
<body bgcolor="#FFFFFF" text="#000000" link="#CCCCCC" vlink="#CCCCCC"
➥alink="#999999" background="wc-bak.gif">
```

Rezultatul acestui efect poate fi văzut în Figura 33.7.

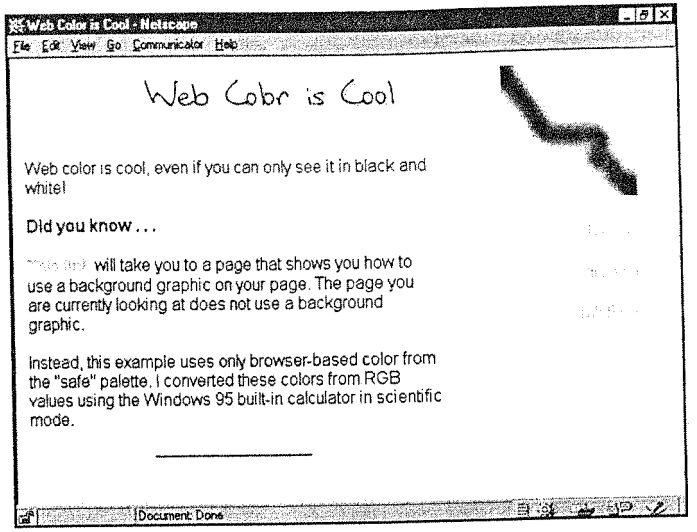


FIGURA 33.6. *Rezultatele în nuanțe de gri ale atributelor pentru legătura creată de mine.*

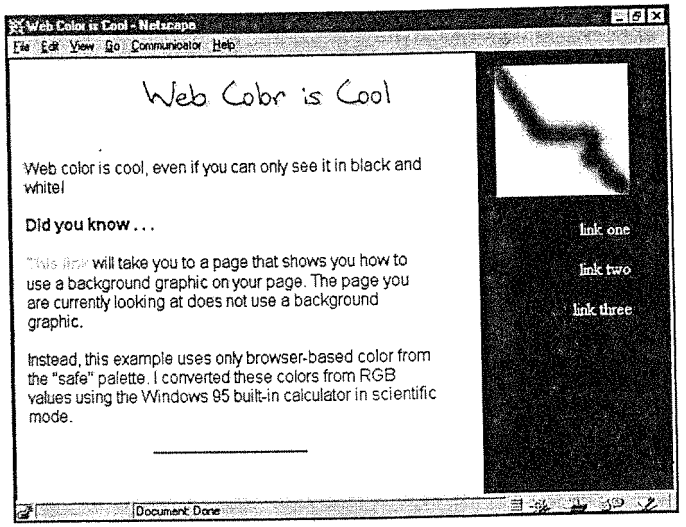


FIGURA 33.7. *Adăugarea unui fundal grafic la pagina prezentată în figura 33.6.*

Variante la hexazecimale

S-ar putea ca unii dintre dumneavoastră să fi văzut atribute HTML de culoare care folosesc numele culorilor în locul valorilor hexazecimale, după cum urmează:

```
<body bgcolor="red" text="white" link="yellow" vlink="green" alink="white">
```

Această metodă a fost introdusă de către browserul Microsoft, Internet Explorer și este compatibilă cu unele versiuni ale altor browsere. Vă sugerez să evitați acest procedeu. Nu este o metodă nici standard, nici stabilă de asigurare a acurateții culorilor pentru platforme și browsere multiple. Are de asemenea limite, și dacă doriți să folosiți culori din paleta

protejată de 216 culori, veți avea în final combinații de nume ale culorilor și valori hexazecimale. Rezultatul este aspectul inconsecvent și ca urmare neprofesional al sitului.

Folosirea culorilor neprotejate

Deși am recomandat folosirea paletei protejate oricând este posibil, acest lucru poate fi restrictiv, datorită faptului că oferă atât de puține culori. Designerii ar putea deveni pe bună dreptate nemulțumiți, așa că următoarele situații permit abaterea de la folosirea paletei protejate:

- Știți destul de bine că publicul dumneavoastră posedă o tehnologie sofisticată de utilizator final, cum este adeseori cazul aplicațiilor din intranet. Când cunoașteți tipul calculatoarelor, browserelor și al monitoarelor pe care le folosește majoritatea publicului, aveți mai multă flexibilitate în alegerea culorii. Dacă majoritatea persoanelor accesează paginile dumneavoastră cu browsere și monitoare sofisticate, puteți folosi culorile după cum credeți de cuviință!
- Dacă cunoașteți mai puțin publicul, dar totuși doriți să folosiți culori neprotejate, testați culorile în cazul intercalării nuanțelor la rezoluții mici. Pentru aceasta, stabiliți rezoluția monitorului la valoarea de 256 de culori. Dacă culorile *se intercalează* (se trece în culoarea cea mai apropiată din paleta sistem) s-ar putea să constatați că o culoare galben dulce se transformă într-o culoare strălucitoare! Acest lucru nu-i va plăcea clientului dumneavoastră. Ideal ar fi să testați rezultatul muncii dumneavoastră pe mai multe calculatoare, la o gamă variată de rezoluții, pentru diferite browsere. Dacă rezultatele sunt suficient de stabile și vă satisfac gusturile, puteți avea o doză de încredere că aspectul culorilor va fi plăcut.

Culorile browserului

Pregătirea și folosirea culorilor folosind metodele descrise în acest capitol vă oferă avantaje în proiectare, prin furnizarea unei baze solide în domeniul noțiunilor fundamentale privind teoria culorilor în Web. Prin înțelegerea unor procedee simple de gestionare a culorii și a modului de folosire a limbajului HTML pentru controlul lor, puteți crea situri bogate, pline de culoare, care radiază energie și au un aspect dinamic, *fără* să utilizeze grafică. Dacă folosiți numai culorile browserului, nu este nevoie ca browserul să interogheze serverul și oferiți vizitatorului timpi de încărcare mai mici, fără a sacrifica estetica aspectului.

Ar trebui ca designerii să folosească cât mai mult posibil culorile browserului. Acest lucru poate fi făcut prin crearea fundalurilor și a culorilor pentru legături, folosind procedeele simple din acest capitol. Pe măsură ce deveniți mai stăpân pe limbajul HTML, veți învăța cum să adăugați culoarea pentru fundalul celulelor tabelelor sau diferite atribute de culoare disponibile în cazul foilor de stil în cascadă. Ideea de bază este de a vă ajuta să folosiți culorile și să renunțați la imagini, deoarece acestea din urmă necesită timp pentru descărcare.

Aceasta nu înseamnă că grafica nu va avea un rol de jucat în cadrul proiectului dumneavoastră – bineînțeles că va avea! Dar în condițiile planificării corespunzătoare și utilizării oportune a culorii protejate, puteți adăuga elemente grafice siturilor dumneavoatră interesante deja, acordând mai puțină atenție timpului de încărcare. Culorile browserului vă lasă libertatea să folosiți disponibilul de fișiere descărcabile cu grafică de înaltă calitate, pentru cazurile în care aveți într-adevăr nevoie. Această metodă este o cale eficientă de a aduce munca dumneavoastră la un nivel profesional.

SUGESTIE

Utilizatorii de Macintosh și PC vor aprecia, cu siguranță, articolul color, ușor de urmărit din situl *MacUser*, `http://www.zdnet.com/macuser/mu_0597/bob/webcolor.html`, apărut în mai '97.

Tehnologia grafică în Web

Nu există scuză pentru gândirea limitată și aspectul neplăcut. Afirmația că grafica unui sit Web trebuie să aibă dimensiuni reduse și să fie mai puțin atractivă vizual reprezintă un mit. Nu, nu vreau să spun că trebuie să împovărați vizitatorii cu descărcări de o durată imposibilă. De fapt, eu încurajez toți designerii să învețe să proiecteze grafica astfel încât siturile lor să permită încărcarea rapidă și să arate bine.

Figura 33.8 prezintă un exemplu de experiență tehnică combinată cu abilități de proiectare. Deși imaginea Paperboy ocupă jumătate din lățimea ecranului și toată lungimea, această pagină se încarcă instantaneu! Designerii acestei pagini au știut cum să-și imagineze proiectul și să aplice tehnologia învățată, ca de exemplu reducerea paletei și machetarea folosind tabelele, pentru a-l contura, a-l face mai interesant și original.

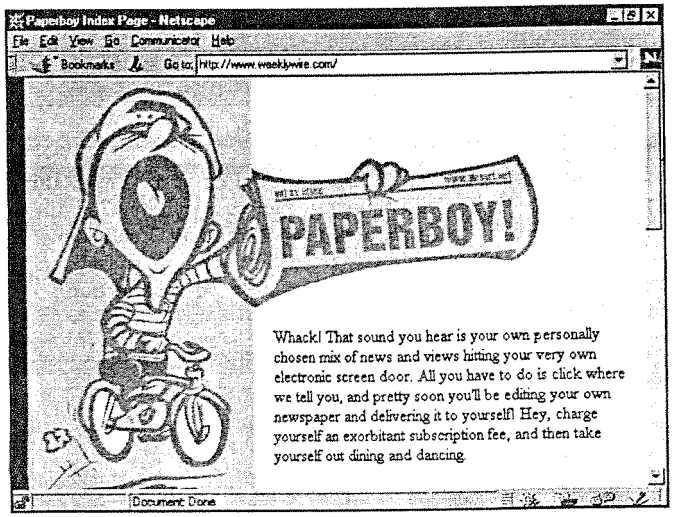


FIGURA 33.8. *Buna planificare combinată cu procede grafice pot garanta rezultate de calitate.*

Evaluarea abilității – graficianul profesionist

Unii dintre cei care citesc acest capitol au deja o pregătire de bază în domeniul graficii pe calculator. Au avantajul de a fi posesorii multor abilități; de asemenea sunt la curent cu materialele și resursele disponibile și știu cum se folosesc multe dintre programele de care este nevoie pentru proiectarea în Web.

Dacă sunteți un grafician tradiționalist, pericolul este reprezentat de incapacitatea de a vă adapta modul de gândire la mediul Web de mică rezoluție. Dacă puteți să vă adaptați modul de gândire, succesul vă este aproape sigur garantat pentru că în acest mediu pot fi exprimate abilitățile, cunoașterea și stilul personal pentru perfecționarea cărora ați muncit mult și bine. Dacă nu studiați totuși modul de readaptare a gândirii, la sfârșit veți avea o

grafică cu timpi de încărcare prea mari sau un proiect care nu se încadrează în restricțiile ecranului calculatorului și, ca rezultat, frustrarea dumneavoastră va fi foarte mare.

Un grafician înțelept, interesat de proiectarea în Web, va trebui să se resemneze – da, veți sacrifica rezoluția înaltă, paletele cu spectru larg și proiectarea orientată pe orizontală, în majoritatea cazurilor. Pe de altă parte, abilitățile dumneavoastră în proiectare vor fi stimulate de faptul că limitările vă forțează uneori să veniți cu soluții creative, inovative. Designerii nou veniți în domeniu trebuie să-și păstreze mintea deschisă, să citească cât mai mult și să exerseze procedeele.

Evaluarea abilității – profesionist sau începător în Web

Alți cititori vor să devină creatori de pagini Web sau lucrează deja în domeniul designului, dar nu au pregătirea necesară sau aceasta este insuficientă. Pentru aceștia, provocarea este să învețe metodele simultan cu acumularea deprinderilor – o sarcină nu prea ușoară! Din fericire, resursele sunt bogate și, cu puțină motivare și multă practică, pot elabora metode foarte bune de a răspunde la provocare.

Totuși, unii sunt total inhibați sau pur și simplu neinteresați de elementele grafice ale proiectării pentru Web. Dacă sunteți una dintre aceste persoane și sunteți responsabil cu construirea siturilor Web, puteți angaja cu ora sau permanent designeri, fotografi și ilustratori, pentru a vă ajuta să vă îndepliniți scopurile. Pentru a vedea cine vă oferă servicii de proiectare pentru grafică de înaltă calitate, vizitați `http://www.portfolios.com/graphic_designers/`.

Instrumente grafice

Dacă ați încercat vreodată să bateți un cui cu șurubelnița, atunci știți că instrumentul potrivit este pur și simplu o necesitate pentru activitate. Unii dintre studenții mei se plâng constant că prețurile instrumentelor Web sunt mari. Realitatea este că, pe termen lung, cei 1200-1500 $ pe care-i veți plăti pentru Photoshop și DeBabelizer reprezintă puțin în comparație cu rezultatele de înaltă calitate pe care le doriți – și se cer – pentru a fi competițional în acest domeniu.

Concepția grafică în Web poate fi considerată ca o știință în sine, dar cerințele actuale pentru a face bine treaba pot fi simplificate la câteva programe. Deși în acest capitol voi menționa câteva componente software pentru grafică, nu uitați că veți avea nevoie de suficient RAM, o placă grafică performantă și un monitor video bun. O necesitate pentru un proiectant Web serios este un scanner color cu pat, precum și o unitate CD-ROM de mare viteză, deoarece multe din materialele sursă și efectele speciale apar pe CD-ROM-uri.

Software pentru grafică

Următoarele programe software pot fi folosite pentru grafica pe Web:

Photoshop: Dacă este pregătit, un proiectant de pagini Web poate învăța cum să folosească produsul Photoshop pentru a face aproape orice – de la crearea și până la modificarea și optimizarea graficii. Mai mult, Photoshop este acceptat în condiții foarte bune în ceea ce privește aplicațiile plug-in și speciale ale firmelor terțe (vezi Figura 33.9). Este disponibil atât pentru platformele Macintosh, cât și pentru cele PC. Prețul estimativ este de 895,00 $ pentru licența standard. Nu există totuși înlocuitor al acestui program puternic, pentru un proiectant serios.

Paint Shop Pro: Îmi place pentru că este primul program bazat pe Windows pentru proiectarea paginilor, destinat realizării graficii atractive cu cheltuieli

minime. Produsul este shareware și poate fi descărcat de la adresa `http://www.jasc.com/`; înregistrarea este 69,00 $.

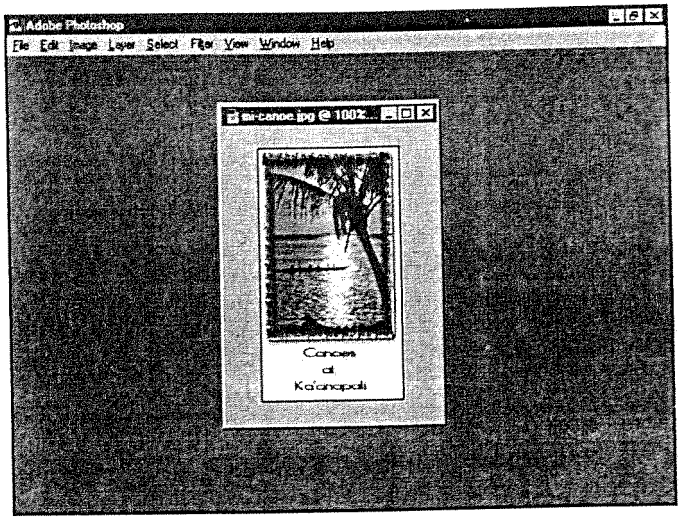


FIGURA 33.9
Aspectul profesional al acestei fotografii a fost creat folosind integral programul Photoshop și programul plug-in Auto F/X.

Corel PhotoPaint: Multe persoane care au deja instalate produsele Corel vor constata că programul Corel PhotoPaint satisface majoritatea cerințelor aplicațiilor Web. Nu îl recomand totuși pentru a fi cumpărat, deoarece Photoshop este un produs superior la aceiași bani, iar Paint Shop Pro este la fel de puternic la un preț mult mai rezonabil. Informații suplimentare privind produsele Corel le puteți găsi la `http://www.corel.com/`.

DeBabelizer Pro: Acest program este un alt concurent de categorie grea pe care proiectanții de pagini Web trebuie să-l ia în considerare pentru setul lor de instrumente. Puterea programului DeBabelizer în aplicațiile Web constă în capacitatea acestuia de a selecta cea mai bună paletă de culori pentru a obține fișiere mici, dar care să păstreze aspectul vizual cât mai plăcut posibil. O proprietate puternică, caracteristică programului DeBabelizer este abilitatea acestuia de a prelucra în pachet sute de fișiere pentru o singură paletă. Această proprietate este utilă pentru cei care creează situri Web mari. Pentru mai multe informații, căutați la `http://www.equilibrium.com/`.

LView Pro: Acest program shareware este un instrument la îndemână pentru conversii grafice PC și realizarea transparenței. Căutați-l la `http://www.lview.com/`.

Materiale sursă și efecte speciale

În afara celor pentru software mai sunt cheltuielile pentru instrumente grafice speciale și materiale sursă. Aceste materiale pot să fie costisitoare, dar, încă o dată, rezultatele sunt pe măsură. Și, gândiți inteligent – dacă lucrați la un anumit proiect, includeți în specificații costurile bunurilor care vă sunt necesare pentru realizarea lui. În acest fel, obțineți plata materialelor și le aveți disponibile pentru proiecte viitoare!

De unde obțineți materialele sursă? Celor aflați la începutul drumului, le recomand cu căldură să viziteze situl Web Internet Baglady, `http://www.dumpsterdive.com/`. Acesta

conține multe lucruri și multe legături la resurse cu mai multe elemente, cu care pot face pe gratis exerciții.

Pentru proiectele profesionale, nu uitați să vă faceți un stoc de fotografii, fișiere clip art și fonturi. Vizitați situl Image Club (`http://www.imageclub.com/`) pentru referințe excelente privind sursele de materialele de calitate. De asemenea, prin vizitarea sitului Photodisc (`http://www.photodisc.com`) veți găsi o sursă de cumpărare a stocurilor de fotografii, fundaluri și legături la alte situri de interes.

Nu uitați vechiul acronim GIGO (Garbage In, Garbage Out – gunoi bagi, gunoi scoți). Acordați timp pentru a vedea ceea ce aveți, stabiliți de ce aveți nevoie și pregătiți-vă să folosiți banii pentru a fi în cea mai favorabilă poziție pentru succes!

Lucrul cu grafica

E-n regulă deci, acum, că știți bine unde sunteți și unde ați dori să mergeți, pot să cobor la elementele fundamentale ale lucrului cu grafica. Un bun punct de plecare este acela pe care eu îl denumesc „cei cinci S ai producerii imaginii":

1. Start – porniți la lucru folosind resurse și unelte de calitate.
2. Scan – scanați imaginea.
3. Size – dimensionați corespunzător imaginea.
4. Select – alegeți tratamente atractive, ca de exemplu suporturi, filtre, chenare și efectele pe margine.
5. Save – salvați fișierele în format potrivit – GIF sau JPG – în funcție de mărimea fișierului și calitatea pe care încercați să o realizați.

Începeți cu calitatea

Sunteți probabil obosit să tot auziți acest lucru, dar el se aplică într-adevăr fiecărui pas al procesului de proiectare! Repet numai pentru a fi sigură că dacă veți rămâne cu o singură idee din acest capitol, aceasta va fi *calitatea contează*. Porniți la drum cu materiale de calitate și șansele de a obține rezultate bune sunt mai mari, chiar și în cazul în care vă optimizați în mod agresiv grafica pentru a termina mai repede.

Figura 33.10 este un exemplu de material fotografic bun. Observați că imaginea este curată și clară. Pe de altă parte, o imagine fotocopiată, neclară va solicita mult mai mult timp și efort, iar rezultatele nu vor depăși un nivel satisfăcător (vezi Figura 33.11).

Câteva sfaturi privind scanarea

Când se scanează grafica pentru a fi folosită în Web, trebuie avute în vedere două lucruri: materialele sursă și rezoluția de scanare.

Materialele sursă se referă la *ce* încercați dumneavoastră să scanați. Fotografiile trebuie să fie clare, curate și de cea mai mare calitate posibilă. Desenele de mână sau materialele tipărite trebuie să fie și ele foarte curate; controlați-le cu grijă pentru a depista petele și praful. Dacă puteți îndepărta ceva *înainte* de scanare, este foarte bine.

Mulți graficieni preferă să scaneze obiecte reale. Aceasta este o parte a designului „organic" – copierea elementelor care există în realitate și folosirea lor pentru a inspira grafica pe calculator. În figura 33.12, am folosit această metodă de utilizare a obictelor de uz cotidian prin scanarea câtorva obiecte luate de pe biroul meu. Folosiți-vă imaginația! Singurul lucru de care trebuie să aveți grijă este să vă asigurați că suprafața de scanare are aspect plăcut și este curată.

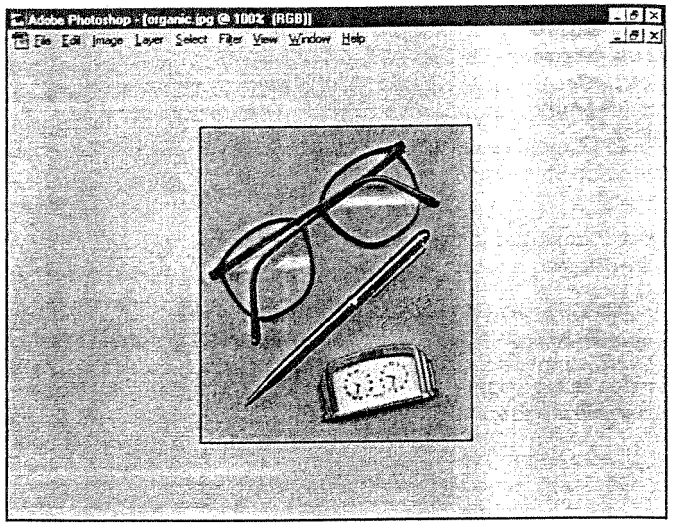
FIGURA 33.10. *Această fotografie a autoarei este pregătită pentru scanare.*

FIGURA 33.11. *O fotocopie boțită și veche necesită efort mult mai mare pentru a fi pregătită.*

Rezoluția de scanare pentru Web poate fi mare la început, dar trebuie să fie în final de 72 dpi (puncte pe inci). Aud murmurele adepților graficii tipărite când scriu acest rând – da, este cea mai mică rezoluție utilizată pentru imprimare. Vă reamintesc totuși că aici nu este vorba despre mijloacele de informare tipărite. În grafica pe calculator, nu aveți nevoie de rezoluție mare pentru a face o imagine *să pară* că are rezoluție mare.

Puteți alege fie să scanați la 72 dpi, fie să scanați la o rezoluție mai mare, ca de exemplu 300 dpi, și apoi să reduceți rezoluția. Am observat că uneori este posibil să se obțină rezultate mai bune dacă se scanează mai întâi la 300 dpi și se reduce apoi rezoluția la 72 dpi. În alte cazuri, acest lucru nu este valabil. Diferența de calitate este adesea mică și nu va fi în mod normal sesizată când se optimizează o imagine pentru a fi folosită în Web.

FIGURA 33.12. *Am luat câteva obiecte tridimensionale de pe biroul meu și le-am scanat direct în Photoshop.*



Personal, nu am găsit o modalitate care să se dovedească a fi optimă, așa că trebuie să aflați care dintre ele funcționează cel mai bine cu tipul dumneavoastră de scanner și este pe măsura sensibilității dumneavoastră. În orice caz, rezultatul final este 72 dpi.

Dimensionarea imaginilor

Asigurați-vă că dimensiunile stricte ale elementelor grafice sunt potrivite din punct de vedere al proporțiilor pentru media Wes. Nu mă refer la mărimea imaginii în kiloocteți, ci la lățimea și înălțimea imaginii. Nu voi discuta nici despre grafica de fundal, prezentată în Capitolul 32, „Navigarea și machetarea". Aici este vorba despre elementele grafice de dimensiuni mici și piesele de detaliu, precum butoanele și liniile despărțitoare.

Amintiți-vă iarăși că lucrați pentru ecrane standard de calculator, cu o rezoluție obișnuită de 640x480 pixeli. Aceasta înseamnă că, pentru a păstra proiectul dumneavoastră în limitele acestei rezoluții, părțile componente trebuie să fie de fapt mai mici, în special în cazurile în care lățimea este o problemă. De fapt, spațiul vizibil este mai mic decât rezoluția de 640x480, deoarece browserul Web ocupă o parte din acest spațiu.

Nu este bine să folosiți dispunerea pe orizontală, decât dacă ați gândit totul în amănunt sau aveți un motiv important pentru aceasta. Când sunt plasate pe orizontală, imaginile trebuie să fie mai mici decât maximul absolut de 600 de pixeli.

Lungimea elementelor grafice depinde de macheta paginii respective. De regulă, este bine ca lungimea imaginilor să fie aproximativ cât înălțimea ecranului. Cu alte cuvinte, doresc să văd întreaga imagine fără să fie nevoie să apelez la defilare.

În sfârșit, de o importanță deosebită este păstrarea proporțiilor față de alte elemente ale paginii. Trebuie să realizați un echilibru agreabil între text, alte imagini și imaginea respectivă. Bineînțeles că mărimea poate varia în anumite limite; verificați dacă aspectul final nu strică proporțiile. În Figura 33.13 puteți vedea o capcană obișnuită care poate fi evitată printr-o proiectare bine echilibrată.

FIGURA 33.13. *Un antet grafic prea mare scoate din ecran meniul de dedesubt.*

Selectarea prelucrărilor grafice

Adăugarea de surse de lumină, filtre speciale, umbre proiectate, suporturi sau efecte de margine pot da viață și dimensiune graficii dumneavoastră. Aceste efecte fac de asemenea ca situl dumneavoastră să arate profesional.

Unele dintre metodele pe care le voi prezenta în capitolele viitoare sunt native în Photoshop, care are multe filtre excelente și tehnici de iluminare pentru a da viață imaginilor dumneavoastră. Pentru graficienii foarte serioși recomand programul Kai's Power Tools; acest pachet de programe conține unele accesorii plug-in care măresc posibilitățile dumneavoastră de proiectare. Sunt necesare de asemenea accesorii plug-in pentru efecte de margine. Pentru acestea, prefer setul furnizat de Auto F/X (`http://www.autofx.com/`).

Figura 33.14 arată cum umbra proiectată conferă textului un efect atractiv. Am folosit unul dintre efectele de margine din Auto F/X pentru a obține un aspect sofisticat pentru această fotografie.

Salvarea corespunzătoare a fișierelor

Acest lucru este unul din cei mai critici pași în optimizarea graficii Web. Modul de salvare al fișierului are influență enormă asupra mărimii sale în kiloocteți și asupra aspectului. Sunt două tipuri principale de fișiere folosite în Web: GIF și JPG (pronunțat și JPEG).

Când se folosește unul în detrimentul celuilalt? Răspunsul este destul de simplu, bazat pe modul de lucru al GIF și JPEG cu paletele de culori și comprimarea. Regula de bază este să folosiți GIF pentru grafica simplă și JPEG pentru fotografii și grafică în care este nevoie de rezoluție mare și fișiere mici.

Formatul GIF

Formatul GIF a fost creat pentru elementele grafice de dimensiuni mici. Modul de lucru se bazează pe reducerea paletei de culori a graficii la maxim 256 de culori. Adesea, ceea ce percepeți dumneavoastră ca fiind o culoare sunt de fapt mai multe; de exemplu, galbenul este compus din mai multe nuanțe sau, în cazul culorilor digitale, din pixeli de culori individuale. Tehnologia GIF reduce numărul de culori folosind *intercalarea* pentru cele care nu se găsesc în paleta GIF standard de 256 de culori.

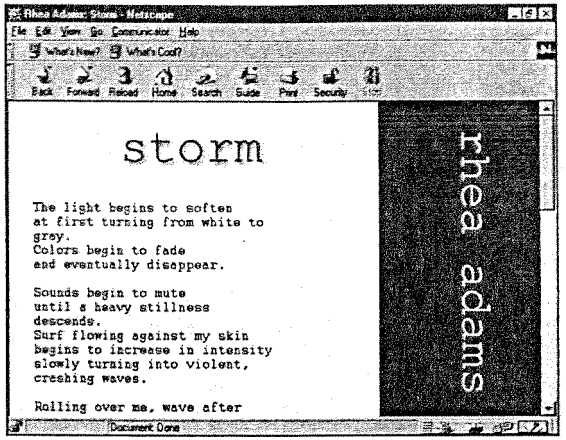


FIGURA 33.14. *Umbra proiectată conferă dimensiune antetului acestei pagini.*

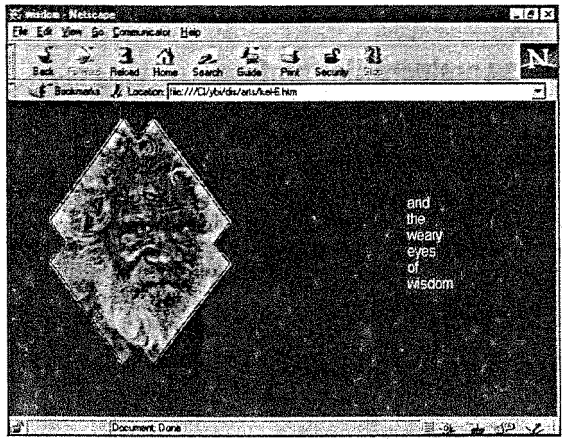


FIGURA 33.15. *Efectele de margine din jurul fotografiei adaugă elemente sofisticate de proiectare unei pagini de altfel simple.*

Formatul JPG

Formatul JPG (scris și JPEG) folosește o tehnologie complet diferită, denumită *comprimare cu pierderi*, pentru a produce fișiere de dimensiuni mai mici. Forța unică a formatului JPG constă în faptul că se face mai puțină intercalare, astfel încât puteți obține o culoare mai apropiată de realitate pentru o dimensiune mai mică a fișierului.

Fișierele GIF, datorită intercalării, sunt mai bune pentru grafica cu mai puține culori, gradare mai mică a culorii (sunt renumite exemplele nereușite de fișiere GIF pentru imaginile care reprezintă cerul), sau mai puține variații de lumină (ca de exemplu stilurile metalic și 3-D).

Fișierele JPG se utilizează cel mai bine pentru fotografii, în special ale cerului sau reprezentând oameni, și pentru grafică de înaltă rezoluție, cu lumină variabilă.

Alte tehnici grafice

Alte tehnici grafice vă ajută să maximizați grafica simultan cu minimizarea fișierelor și a problemelor inerente pentru ecranul calculatorului. Printre aceste procedee se numără randarea progresivă și transparența.

Fișiere GIF întrețesute și fișiere JPG progresive

Randarea progresivă reprezintă apariția gradată a graficii, o metodă utilă pentru a păstra atenția publicului pe durata încărcării. Imaginea apare mai întâi „încețoșată" și apoi, în mod progresiv, devine mai clară.

Există două metode de randare progresivă în Web. Cea mai populară și mai eficientă este folosirea fișierelor GIF întrețesute. Procedeul poate fi aplicat oricărui fișier GIF și, de obicei, în programele dumneavoastră grafice sau instrumente plug-in există opțiuni care vă ajută să vă desfășurați munca de creație.

Fișierele *JPG progresive* reprezintă răspunsul JPEG la întrețesere. Nu puteți face întrețeserea unui fișier JPG, dar există programe care creează ceea ce este cunoscut sub numele de fișier JPG progresiv. De exemplu, Photoshop 4.0 are opțiunea de randare progresivă pentru JPG. Personal, nu sunt prea încântată de fișierele JPG progresive, deoarece cred că tehnologia este oarecum incipientă, iar calitatea nu este obținută în mod consecvent. În modul de funcționare standard, fișierele JPG neprogresive defilează în loc să randeze progresiv, dar în cazul în care ați realizat optimizarea, machetarea și codarea minuțioasă, acest lucru nu constituie o problemă, în special deoarece rezultatele sunt sub o formă prezentabilă.

Transparența

Transparența vă permite crearea unor fundaluri cu texturi și „imprimarea" unui element grafic peste acestea fără a întrerupe modelul fundalului. Acest lucru presupune crearea unei culori „transparente" în grafică. Acea culoare transparentă dispare când este așezată pe o pagină cu textură. Acest procedeu cere puțină răbdare și timp pentru a fi învățat și poate fi realizat cu instrumente shareware, ca de exemplu Lview Pro, sau cu Photoshop.

Țineți minte că transparența poate fi realizată numai cu fișiere GIF, așa că, dacă doriți să realizați transparența unei fotografii de calitate mare, nu trebuie să folosiți texturi ca fundal al unei fotografii pretențioase.

Imaginile și limbajul HTML

Imaginile sunt poziționate în pagină folosind limbajul HTML. Aveți la îndemână mai multe atribute disponibile în etichetele imaginilor, care permit controlul maxim asupra modului de așezare a imaginii în pagină. Mai mult, folosirea tabelelelor pentru controlul graficii reprezintă o metodă foarte eficientă de control al machetării. Pentru amănunte referitoare la machetare, nu uitați să citiți Capitolul 34, „Folosirea spațiului, a formei și a tipului caracterelor".

Pentru atingerea scopurilor dumneavoastră imediate, vă voi prezenta eticheta standard pentru imagini și atributele acesteia. Eticheta cu atribute obișnuite a unei imagini arată astfel:

```
<img src="header.gif" alt="example header graphic" width=300 height=50>
```

Voi examina părțile separate ale acestui exemplu înainte de a trece la atribute mai complexe.

- `img`: Abrevierea provine de la *image* (imagine); atributul `img` avertizează browserul și serverul că va fi plasată o imagine în acest punct al structurii HTML.
- `src=x`: Este prescurtarea pentru *source* (sursă); atributul `src` definește adresa imaginii pe server.

■ **`alt=x`:** Atributul `alt` permite ca un text alternativ, care poate înlocui imaginea, să fie introdus în șirul de atribute ale imaginii. Acest atribut este folositor în special pentru persoanele care folosesc browsere numai pentru text sau au anulat opțiunea pentru imagini; acest atribut vă poate ajuta la identificarea imaginii, deoarece apare textul explicativ când mouse-ul trece peste imagine (vezi Figura 33.16).

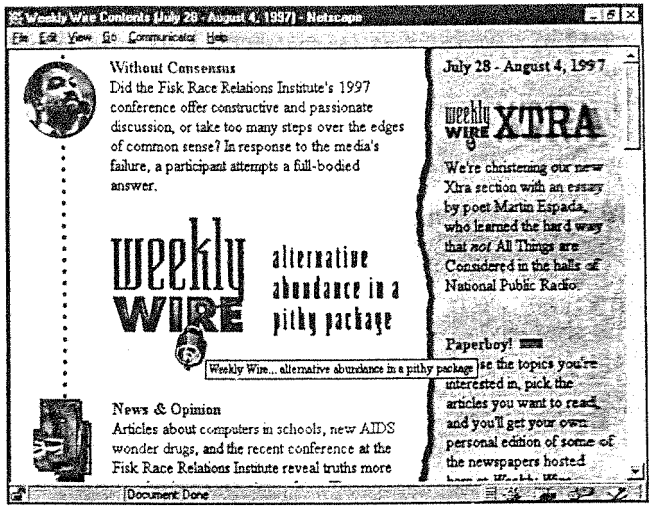


FIGURA 33.16. *Atributul* `alt` *definește textul comentariului pop-up când mouse-ul trece peste imagine.*

■ **`width=x`:** Aceasta este lățimea imaginii. Specificarea lățimii permite browserului să pregătească pentru imagine spațiul adecvat pe orizontală.
■ **`height=x`:** Înălțimea imaginii se referă la dimensiunea acesteia pe verticală. Ca și în cazul lățimii, stabilirea înălțimii ușurează funcționarea browserului.

Alte atribute obișnuite care pot fi folosite în cadrul etichetelor pentru imagini sunt:

■ **`border=x`:** Dacă i se dă atributului valoarea 1 sau mai mult, în jurul imaginii va apărea un chenar. Însă mulți proiectanți se feresc de acest atribut, ceea ce vă recomand să faceți și dumneavoastră. Imaginile arată mai profesional fără un chenar care le limitează. Pentru a elimina chenarul, stabiliți valoarea 0 pentru acesta.
■ **`hspace=x`:** Atributul definește spațiul orizontal, adică spațiul aflat între imagine și orice text sau imagine de-a lungul liniei orizontale.
■ **`vspace=x`:** Spațiul vertical, analog celui orizontal, este spațiul dintre imagine și orice altă informație pe axa verticală.
■ **`align=x`:** Alinierea permite ca o imagine să fie așezată diferit în cadrul spațiului alocat imaginii. Cele mai uzuale valori pentru aliniere sunt:
  `right`: Această valoare aliniază imaginea la marginea din dreapta.
  `left`: Această valoare aliniază imaginea la marginea din stânga.

OBSERVAȚIE

Sunt și alte opțiuni pentru aliniere, despre care puteți citi în alte capitole ale cărții, cum ar fi Capitolul 11, „Adăugarea imaginilor pe pagina dumneavoastră Web" și Capitolul 7, „Alinierea și formatarea textului".

Rezumat

În acest capitol ați învățat cum să combinați paletele, să lucrați cu culorile protejate și să folosiți limbajul HTML pentru a oferi graficii cea mai bună poziție în pagină. Aceste procedee vă oferă avantaje când este vorba despre proiectarea siturilor Web, deoarece sunt adesea neglijate. Faptul că în Web domnește haosul se datorează în parte neacordării atenției cuvenite acestor metode!

Următorul capitol tratează mai amănunțit problemele de aliniere, dar și modul de interacțiune al imaginilor și textului pe pagină. Grafica reprezintă fundamentul proiectării vizuale a paginii Web. Aceasta oferă posibilități de creștere a experienței nu numai prin adăugarea reprezentărilor pictografice, dar și a culorii, formei și a tipografiei. Citiți mai departe pentru a afla mai multe despre modul cum să folosiți cunoștințele noi privind grafica în contextul acestor probleme de proiectare importante.

Capitolul 34
Folosirea spațiului, a formei și a tipului caracterelor

de Molly E. Holzschlag

În acest capitol:

- **Am nevoie de spațiu! 516**
- **Spațiul alb 518**
- **Controlul machetării cu tabele și cadre 528**
- **Forma 530**
- **Tipografia 532**
- **Culoarea fonturilor 536**
- **Alte considerații tipografice 536**

Spațiul, forma și tipul caracterelor sunt elemente esențiale de proiectare. Acest capitol vă va ajuta să învățați unele dintre procedeele HTML necesare pentru realizarea lor, dându-vă în același timp o imagine de ansamblu a aspectelor teoretice care se află în spatele acestor procedee. Aceste informații vă ajută să folosiți limbajul HTML pentru a concepe situri Web care dau dovadă de stil și au un impact deosebit.

Îmi displace profund să mă aflu într-un spațiu limitat. Nu există nimic mai frustrant decât spațiul mic al unei pagini Web care este așa de aglomerat că îți provoacă claustrofobie. Acest lucru îmi amintește de casa bunicii – în orice direcție te mișcai, te ciocneai de un bibelou sau de un obiect cu valoare sentimentală. Spațiul este esențial.

Și forma lucrurilor este esențială. Forma ajută la eliberarea spațiului și stârnește interesul vizual pentru acea pagină. Ați observat probabil că majoritatea paginilor Web sunt pline de dreptunghiuri. Ei bine, ajunge! Nu vreau să spun că dreptunghiurile sunt chiar indezirabile sau complet inutile, dar în acest capitol vă voi arăta nu numai cum să eliberați spațiul disponibil, ci și de ce trebuie să introduceți alte forme în situl dumneavoastră.

Alt element al proiectării Web care stimulează interesul vizual și ajută la crearea spațiului și a formei este tipul caracterelor. Modificările limbajului HTML au început să ofere proiectanților controlul deplin asupra tipului caracterelor, așa că veți învăța ceva despre modul de folosire al tipografiei în Web.

Am nevoie de spațiu!

Priviți ecranul calculatorului dumneavoastră. Ce formă are? E un dreptunghi. Din punct de vedere tehnic, este un dreptunghi, cu anumite dimensiuni pe orizontală și verticală, în funcție de numărul de pixeli. În funcție de tipul monitorului și placa grafică pe care le aveți, valoarea vizuală a obiectului dumneavoastră poate fi 640x480, 800x600 sau mai mare. Această valoare este numită *rezoluție*.

Rezoluția este o problemă căreia designerii talentați îi acordă mare atenție, pentru că aceasta determină cât spațiu aveți de fapt la dispoziție. Se cunoaște că majoritatea persoanelor au sau văd siturile Web la o rezoluție de 640x480, care este cea mai mică rezoluție pentru un monitor obișnuit. Vă recomand foarte mult să folosiți această rezoluție ca punct de plecare în proiectarea spațiului.

În termeni simpli, acest lucru înseamnă că aveți maxim 640 de pixeli în lățime și 480 de pixeli în lungime pentru fiecare ecran cu informații. Cele mai multe pagini Web defilează, astfel că determinarea lungimii nu este așa de critică, cu excepția cazului în care doriți ca pagina dumneavoastră să fie fixă, cum este cea de deschidere (vezi Figura 34.1). Nu aceeași problemă este cu lățimea. Barele orizontale de defilare sunt efectiv rar folosite, astfel încât trebuie să tratați dimensiunile pe orizontală cu mare atenție. Acest lucru înseamnă că grafica, tabelele și alte procedee pe care le folosiți pentru repartizarea spațiului în pagină în mod sigur vor avea maximum 640 de pixeli lățime.

FIGURA 34.1. *Paginile de deschidere sunt adeseori fixe, fără defilare pe verticală.*

Spun maximum pentru că sunt și alți factori care joacă un rol în deciziile de alocare a spațiului. Deschideți browserul dumneavoastră Web. În funcție de preferințele dumneavoastră, puteți avea oricâte bare cu instrumente deschise, care ocupă spațiu pe verticală. Mai pot fi de asemenea o bară de defilare în dreapta și o bară de stare de-a lungul părții de jos a ecranului (vezi Figura 34.2). Toți acești factori reduc efectiv cantitatea de pixeli cu care puteți lucra.

Pe scurt, aveți un spațiu de maximum 600 de pixeli în lățime și de 400 de pixeli sau mai puțin în lungime, iar faptul că țineți cont de aceste dimensiuni vă dă libertate de mișcare în loc să vă constrângă.

Gândiți-vă la acea cameră mică din casa dumneavoastră, aceea fără ferestre. Ce puteți face pentru ca spațiul să pară mai mare? Ei bine, o metodă este s-o zugrăviți în alb sau altă culoare deschisă. Altă metodă este să scoateți ceva mobilă afară – iar mobila pe care *hotărâți* să o puneți în cameră să fie bine aleasă, deschisă la culoare și mai ușoară. În final puteți adăuga oglinzi sau tablouri care extind orizontul vizual al camerei.

Deschiderea spațiului din pagina dumneavoastră Web este o problemă similară. Folosind una sau toate metodele din multitudinea celor pentru care camera pare mai mare, veți realiza o pagină care are un aspect mai aerisit, mai luminos și mai curat.

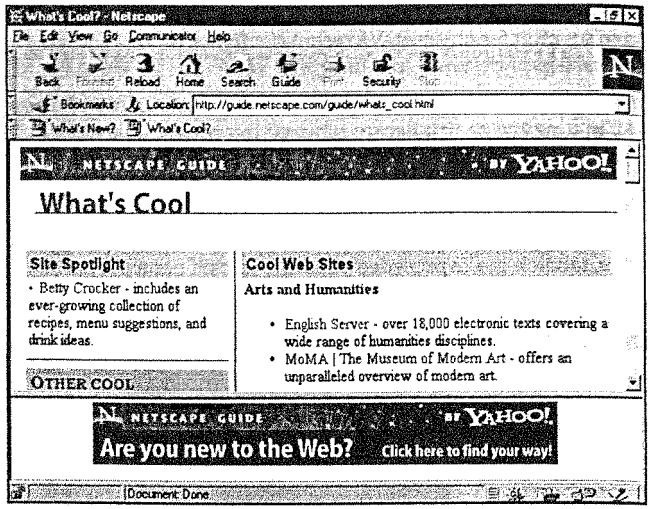


FIGURA 34.2. *Browserul Web Netscape cu toate barele cu instrumente și barele de stare afișate – toate acestea ocupă spațiu.*

Faptul că dumneavoastră cunoașteți restricțiile cu care lucrați *înainte* de a concepe pagina vă eliberează ideile în loc să le limiteze. Știți cu ce lucrați, așa că puteți avea opțiuni mai bune și mai inteligente, îmbunătățind aspectul paginilor și făcând ca vizitatorii dumneavoastră să se simtă mai confortabil.

Spațiul alb

Zona fără text sau grafică de pe paginile unui ziar sau ale unei reviste este denumită *spațiu alb*. Bineînțeles că acest spațiu poate să nu fie alb, dar este spațiu liber. Adesea, ce *nu* este în pagină e la fel de important cu ceea ce este în pagină.

Într-o pagină Web, spațiul alb este imperativ necesar, dar trist este faptul că acest aspect este neglijat adesea. Spațiul alb îndreaptă ochiul către informația care va veni, odihnește ochiul prin spațiul liber de informație și armonizează textul și grafica, făcându-le să arate diferit de camera aglomerată a bunicii.

Următoarele metode HTML vă vor ajuta să introduceți spațiu alb pe pagina dumneavoastră, îmbunătățind experiența vizitatorului. Chiar dacă nu iese imediat în evidență, un sit Web aglomerat transmite tensiune. Nu doriți ca vizitatorul dumneavoastră să plece – scopul este crearea unui mediu care să-i placă! Spațiul ajută la decorarea paginii dumneavoastră într-o manieră atractivă.

Marginile

HTML oferă o metodă standard de a controla marginile: folosirea etichetei `<BLOCKQUOTE>`. Eticheta `<BLOCKQUOTE>` și tovarășa acesteia, `</BLOCKQUOTE>`, poziționate în jurul conținutului unei pagini, creează instantaneu margini și vă oferă prețiosul spațiu alb la stânga și la dreapta conținutului. Iată un exemplu de cod:

```
<html>
<head>
<title>Blockquoting = White Space!</title>
</head>
```



```
<body>
<blockquote>
Duis autem vel eum iriure dolor in hendrerit in vulputate velit esse molestie
consequat, vel illum dolore eu feugiat nulla facilisis at vero eros et accumsan
et iusto odio dignissim qui blandit praesent luptatum zzril delenit augue duis
dolore te feugait nulla facilisi.
<p>
<img src="seattle.jpg">
<p>
Nam liber tempor cum soluta nobis eleifend option congue nihil imperdiet doming id
quod mazim placerat facer possim assum. Accumsan et iusto odio dignissim qui blandit
praesent luptatum zzril delenit augue duis dolore te feugait nulla facilisi.
<p>
</blockquote>
</body>
</html>
```

Figura 34.3 prezintă rezultatele. Comparați această imagine cu Figura 34.4, unde am eliminat marginile. Diferența e evidentă!

OBSERVAȚIE

Eticheta `<BODY>` are două atribute pe care le puteți folosi când proiectați pentru Internet Explorer. Acestea sunt atributele `leftmargin=x` și `topmargin=x`, care vă oferă controlul asupra marginii din stânga și de sus a paginii. De exemplu, dacă doresc să am aproximativ 30 de pixeli de spațiu alb la stânga, iar pagina mea să fie plasată la limita de sus a spațiului de pagină din browser, aș folosi următoarea sintaxă:

```
<body leftmargin=20 topmargin=0>
```

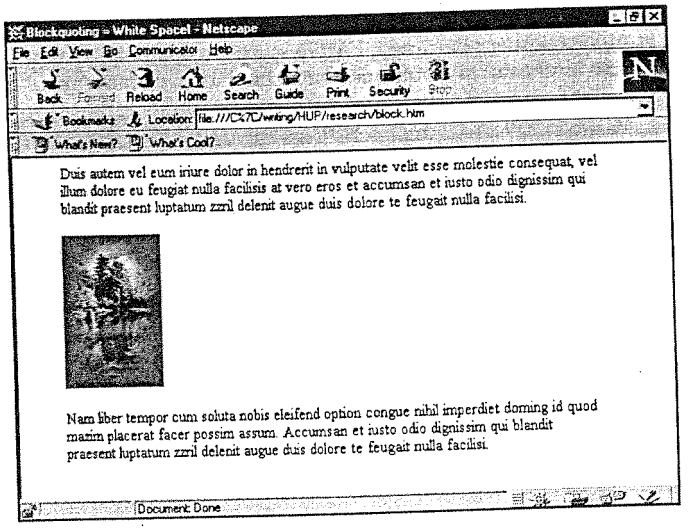


FIGURA 34.3. *Eticheta `<BLOCKQUOTE>` ajută la crearea marginilor, astfel încât textul dumneavoastră nu umple ecranul pe orizontală.*

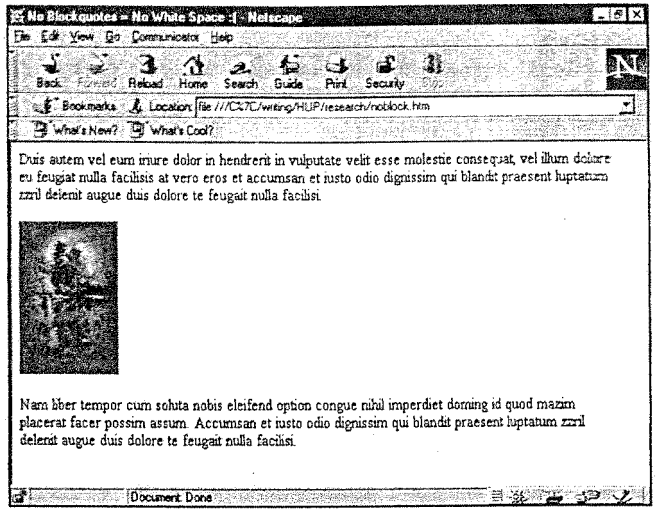


FIGURA 34.4. *Scoateți eticheta* <BLOCKQUOTE> *și nu veți mai avea margini – și nici spațiu efectiv.*

Alinierea

Alinierea reprezintă locul și modul de plasare a textului, imaginilor și obiectelor pe pagină. Există mai multe metode de realizare a alinierii folosind limbajul HTML.

Alinierea textului și a blocurilor de date

Alinierea textului și a imaginilor vă ajută să dați un aspect plăcut paginilor dumneavoastră. Din punct de vedere al proiectării, trebuie să aliniați textul unde arată mai bine, nu să obțineți doar un aspect diferit. De exemplu, corpul textului standard ar trebui să rămână cu familiara aliniere la stânga. Părțile scurte de text pot fi aliniate la dreapta, iar gruparea pe centru a textului și a obiectelor ar trebui făcută cu mare grijă.

ATENȚIE!

Centrarea textului e o practică obișnuită, iar ochiul neantrenat *are impresia* că arată mai bine pe pagină decât textul aliniat după moda veche, la stânga. Motivul pentru care arată mai bine este că centrarea textului creează spațiu alb! Acest spațiu este cel care place ochiului – și nu textul centrat, care este de fapt mult mai dificil de citit în totalitate. Centrarea trebuie rezervată pentru porțiuni specifice, condensate de text, cum sunt subtitlurile. În alte situații, reduceți nevoile de centrare a textului prin apelarea la alte opțiuni de aliniere și creare a spațiului.

Următoarele etichete și atribute HTML vă permit controlul alinierii într-o varietate de moduri:

- **<DIV>...</DIV>:** Eticheta <DIV> sau *division* (diviziune) este una dintre cele mai puternice căi de a împărți blocurile de informație și a le alinia. Conținutul acestei etichete se va conforma unuia dintre următoarele atribute de aliniere:

center: Acest atribut centrează toate informațiile dintre etichetele <DIV> și </DIV>, după cum se vede mai jos:

```
<div align=center>
Duis autem vel eum iriure dolor in hendrerit in vulputate velit esse molestie
consequat, vel illum dolore eu feugiat nulla facilisis at vero eros et accumsan
et iusto odio dignissim qui blandit praesent luptatum zzril delenit augue duis
dolore te feugait nulla facilisi.
<p>
<img src="seattle.jpg">
<p>
</div>
```


Rezultatul este prezentat în figura 34.5.

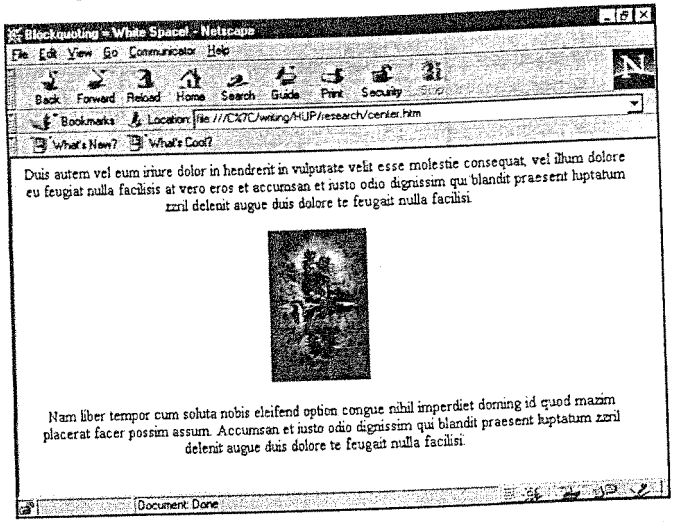


FIGURA 34.5. *Alinierea pe centru folosind eticheta* <DIV>.

left: Toate datele vor fi aliniate la stânga. Alinierea la stânga e prestabilită pe pagină și, cu excepția cazurilor în care sunteți într-un tabel sau într-o etichetă, veți folosi destul de rar acest atribut.

right: Acest atribut aliniază la dreapta datele dumneavoastră:

```
<div align=right>
Duis autem vel eum iriure dolor in hendrerit in vulputate velit esse molestie
consequat, vel illum dolore eu feugiat nulla facilisis at vero eros et accumsan
et iusto odio dignissim qui blandit praesent luptatum zzril delenit augue duis
dolore te feugait nulla facilisi.
<p>
<img src="seattle.jpg">
<p>
</div>
```

Puteți observa în Figura 34.6 alinierea textului și a imaginii la dreapta.

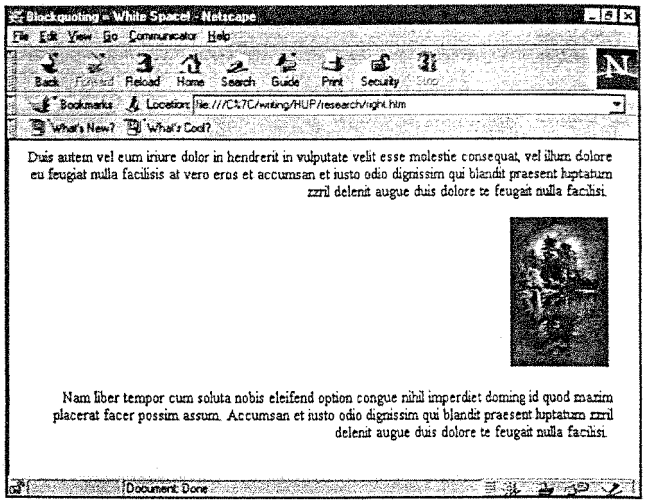
- ** ... :** Dacă folosiți perechea de etichete pentru paragraf cu aceleași atribute din eticheta , realizați același tip de aliniere.

center: Tot textul din următorul exemplu va fi centrat:

```
<p align=right>
Duis autem vel eum iriure dolor in hendrerit in vulputate velit esse molestie
consequat, vel illum dolore eu feugiat nulla facilisis at vero eros et accumsan
```

```
et iusto odio dignissim qui blandit praesent luptatum zzril delenit augue duis
dolore te feugait nulla facilisi.
</p>
```

FIGURA 34.6. *Alinierea la dreapta cu ajutorul etichetei* `<DIV>`.



`left`: Încă o dată, alinierea la stânga este valoarea standard pentru paragrafe, așa că veți utiliza foarte rar sau niciodată acest atribut cu eticheta de paragraf.

`right`: Datele următoare vor fi aliniate la dreapta:

```
<p align=right>
Duis autem vel eum iriure dolor in hendrerit in vulputate velit esse molestie
consequat, vel illum dolore eu feugiat nulla facilisis at vero eros et accumsan
et iusto odio dignissim qui blandit praesent luptatum zzril delenit augue duis
dolore te feugait nulla facilisi.
</p>
```

```
<p align=right>
<img src="seattle.jpg">
</p>
```

SUGESTIE

După cum observați din exemplul precedent de cod, alinierea paragrafelor necesită etichetă de deschidere și de închidere pentru fiecare secvență de date care trebuie aliniată. De aceea, dacă doriți să aliniați mai mult de o secvență pe o pagină, e mai rapid și mai elegant să folosiți opțiunea `<DIV>`.

O metodă obișnuită de centrare a textului este folosirea etichetei `<CENTER>` și a tovarășei acesteia, `</CENTER>`, pentru închidere. Personal, nu vă sfătuiesc să folosiți această metodă, deoarece eticheta `<DIV>` e mai elegantă și compatibilă cu mai multe browsere.

Alinierea imaginilor

Imaginile pot fi aliniate atât pe orizontală, cât și pe verticală. Alinierea imaginilor e făcută chiar în eticheta `<IMG>` cu atributul `align=x`. Sintaxa pentru o imagine aliniată la stânga arată astfel:

```
<img src="my-face.gif" align=left>
```

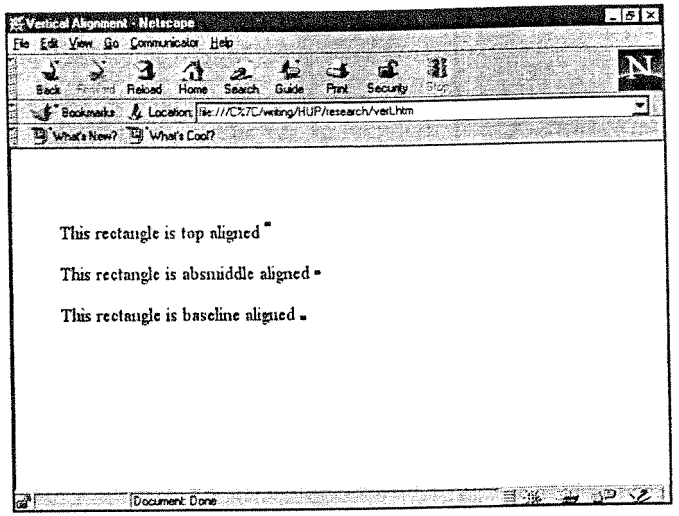

Pentru alinierea pe verticală a imaginilor, folosiți următoarele valori de aliniere (vezi Figura 34.7):

`top`: Textul va fi aliniat cu partea de sus a imaginii.

`absmiddle`: Mijlocul textului va fi aliniat cu mijlocul imaginii.

`baseline`: Partea de jos a textului va fi aliniată cu partea de jos a imaginii.

FIGURA 34.7. *Alinierea pe verticală la lucru.*



SUGESTIE

Atributele `baseline` și `absmiddle` se folosesc mai rar în comparație cu alte atribute prezentate aici. Ele sunt folosite totuși de designeri în cazurile în care sunt necesare poziționări mai precise.

OBSERVAȚIE

Există și multe alte opțiuni pentru alinierea pe orizontală, față de cele prezentate aici, a căror repetare ar fi lipsită de sens. Pentru informații suplimentare asupra tuturor opțiunilor disponibile de aliniere pe orizontală, consultați Anexa A, „HTML 4.0 – Referințe".

Alinierea pe orizontală a imaginilor se bazează pe următoarele atribute (vezi Figura 34.8):

`left`: Acest atribut aliniază imaginile la stânga.

`right`: Acest atribut aliniază imaginile la dreapta.

Imaginile pot fi aliniate cu textul. Puteți avea mai mult control asupra acestei alinieri prin folosirea etichetei `<br clear=x>` și a atributului.

`<br clear=right>`: Folosiți această etichetă după o imagine aliniată la stânga, pentru a forța corpul textului spre dreapta, după cum se vede în Figura 34.9.

`<br clear=left>`: Dacă ați aliniat o imagine la dreapta, această etichetă forțează corpul textului spre stânga imaginii, după cum se vede în Figura 34.10.

`<br clear=all>`: Folosiți această etichetă pentru a anula toate informațiile referitoare la imagine (vezi Figura 34.11).

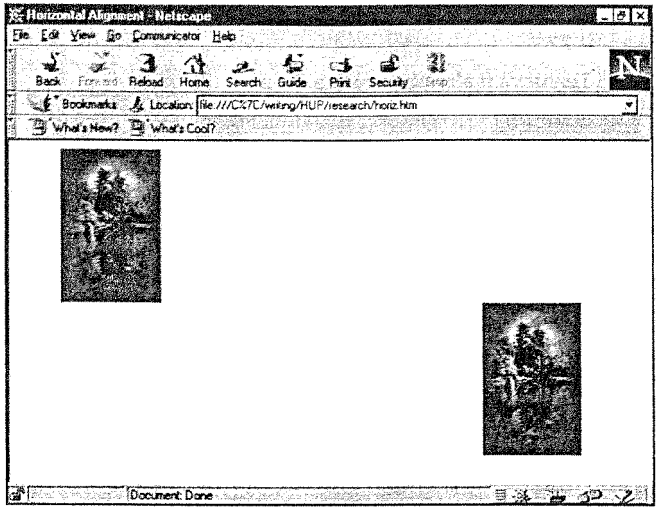


FIGURA 34.8. *Alinierea pe orizontală.*

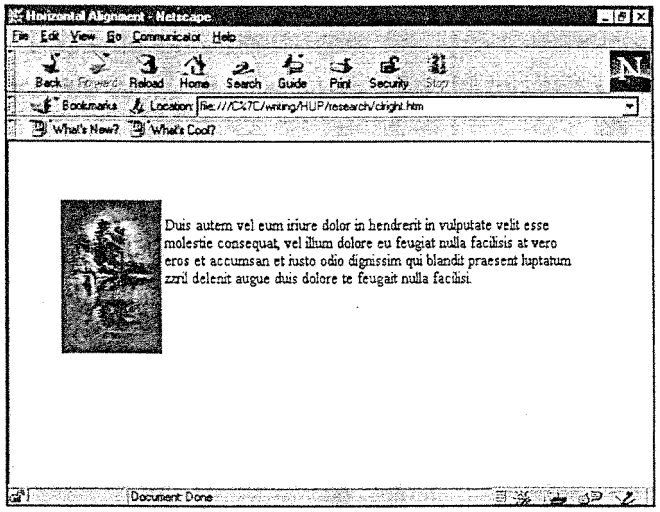


FIGURA 34.9. *Forțarea apariției textului în partea dreaptă a imaginii.*

OBSERVAȚIE

Este important să ținești minte că limbajul standard HTML 4.0 recomandă pentru poziționare folosirea foilor de stil în detrimentul atributelor de aliniere. Acest lucru este totuși susținut în contextul folosirii browserelor care sunt compatibile cu foile de stil. Actualmente, singurele browsere mai cunoscute care sunt compatibile cu foile de stil sunt IE versiunea 3.0 sau ulterioară și Netscape 4.0. Aceste programe sunt folosite de o minoritate a comunității Web, așa că este întotdeauna bine să ținești seama de public înainte de adoptarea limbajului standard HTML 4.0 pentru proiectele Web comerciale.

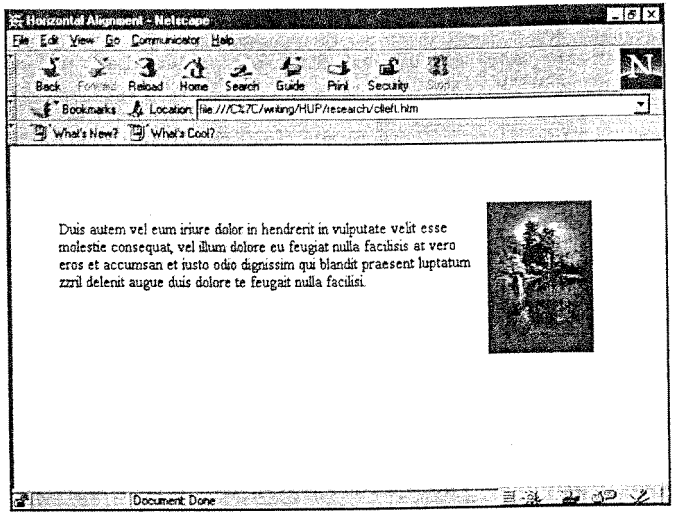


FIGURA 34.10. *Forțarea apariției textului în partea stângă a imaginii.*

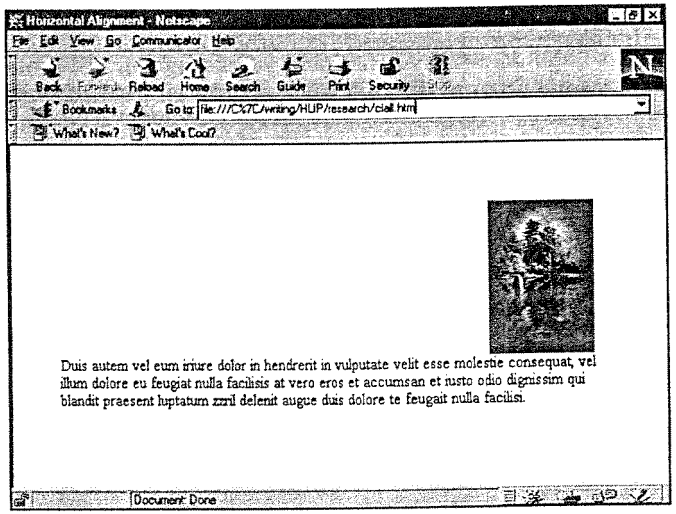


FIGURA 34.11. *Anularea informațiilor.*

Imagini flotante

Există o altă metodă importantă, ceva mai avansată, cu care puteți alinia imaginile. Aceasta este o aliniere mai dinamică, care se bazează pe relația dintre blocurile de text și imaginea din interiorul textului. Procedeul este important în contextul alinierii și spațiului, deoarece reprezintă un instrument puternic de a crea spațiu pe o pagină.

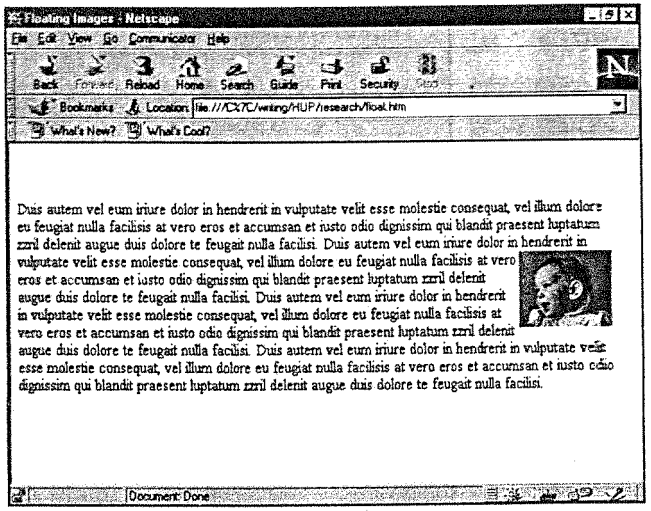
Imaginile flotante sunt create prin eliminarea tuturor pauzelor de după imagine. De exemplu, dacă am o imagine și vreau s-o încorporez într-un paragraf, mai întâi poziționez și aliniez imaginea și apoi poziționez paragraful, fără a introduce nici o pauză între imagine și paragraf:

```
<img src="my-face.gif" align=right>
```

```
Duis autem vel eum iriure dolor in hendrerit in vulputate velit esse molestie
consequat, vel illum dolore eu feugiat nulla facilisis at vero eros et accumsan
et iusto odio dignissim qui blandit praesent luptatum zzril delenit augue duis
dolore te feugait nulla facilisi.
<p>
```

Figura 34.12 prezintă rezultatele. Observați cum textul se înfășoară în jurul imaginii. O astfel de imagine se numește imagine *flotantă*. Cei care au vederea ageră vor observa însă lipsa unui lucru. Ați ghicit – spațiul alb!

FIGURA 34.12. *O imagine flotantă, înainte de a adăuga spațiul alb.*



Puteți adăuga spațiu alb în jurul unei imagini flotante folosind următoarele atribute:

hspace: Acesta este *spațiul orizontal*. Atributul controlează cât spațiu alb este pe orizontală, în partea dreaptă și stângă a imaginii.

vspace: Atributul *spațiu vertical* introduce spațiu alb pe verticală, în partea de sus și de jos a imaginii.

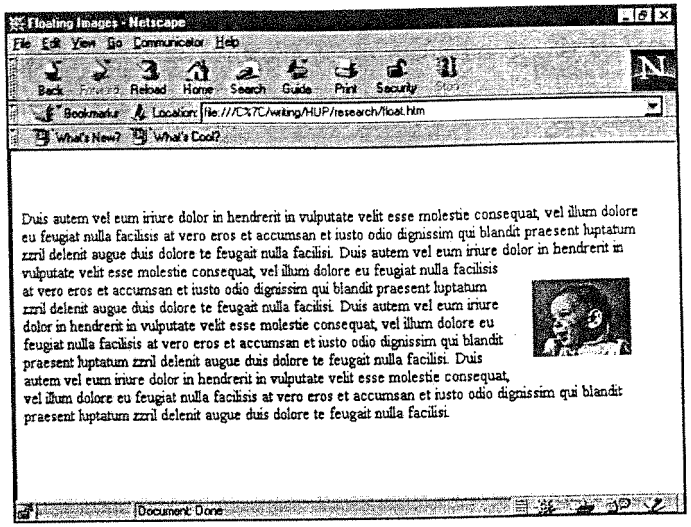
Dacă schimb codul corespunzător imaginii din exemplul precedent, veți observa în Figura 34.13 că am creat suficient spațiu alb în jurul imaginii mele:

```
<img src="my-face.gif" align=right hspace=20 vspace=20>
```

Proporția și proximitatea

O altă problemă importantă referitoare la spațiu este cea referitoare la proporție și proximitate. *Proporția* exprimă cât este de mare o anumită imagine, un anumit obiect sau porțiune de text în comparație cu alte imagini, obiecte sau porțiuni de text din pagină. Proporția poate fi înțeleasă dacă ne gândim la diferite conformații ale corpului uman. Unii au picioare foarte lungi și trunchiul scurt. Unele fețe sunt rotunde, altele alungite. Diversitatea oamenilor este o parte a ceea ce îi face interesanți din punct de vedere estetic, dar în cazul paginilor Web designerii se străduiesc să păstreze echilibrul între părți. Cu alte cuvinte, un proiect bun poate fi comparat cu o conformație uniformă a corpului, sau mai bine zis cu un corp „bine proporționat".

FIGURA 34.13. *O imagine flotantă care folosește atributele* vspace *și* hspace.



Proximitatea înseamnă cât de aproape este un obiect față de celălalt. Dacă cineva are ochii prea apropiați sau depărtați, veți percepe aspectul persoanei respective ca fiind oarecum neobișnuit. Din nou, proporția este aici soluția pentru proiectarea reușită, păstrând o distanță între obiecte aproape constantă, nici prea mică, nici prea mare.

Figura 34.14 prezintă o pagină cu obiecte proporționate defectuos, care sunt necorespunzătoare și din punct de vedere al proximității. Am remediat deficiențele în Figura 34.15, prin redimensionarea și poziționarea imaginilor în mod echilibrat în pagină.

FIGURA 34.14. *Elemente disproporționate și neechilibrate în pagină.*

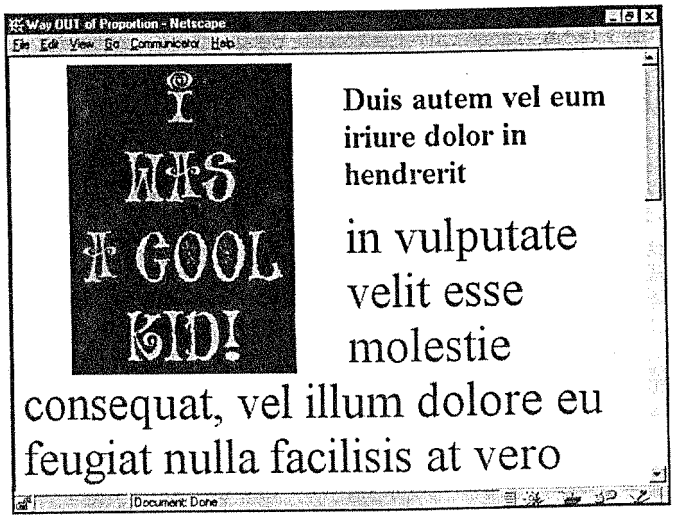


FIGURA 34.15. *Aceeași pagină căreia i s-a aplicat proximitatea și proporția.*

Țineți cont de proximitate și proporție în proiectarea siturilor. Cu cât sunt mai echilibrate fiecare, cu atât va fi mai natural spațiul sitului și cu atât mai satisfăcut veți fi – dumneavoastră și vizitatorii sitului dumneavoastră – de rezultate.

Controlul machetării cu tabele și cadre

Când lucrați cu tabele și cadre, puteți să așezați paginile, să controlați imaginile și textul și, în final, să puneți în ordine spațiul, proporția și proximitatea elementelor paginii Web. Aplicarea machetei și a cadrelor implică folosirea extensivă a limbajului HTML, așa că voi prezenta câte un caz simplu pentru a vă stimula apetitul. Apoi, vă recomand să recapitulați Partea a-II-a a acestei cărți, „Elemente de structură și navigație ale HTML-ului 4.0". De asemenea, Capitolul 20, „Crearea cadrelor", prezintă detaliile folosirii efective a cadrelor.

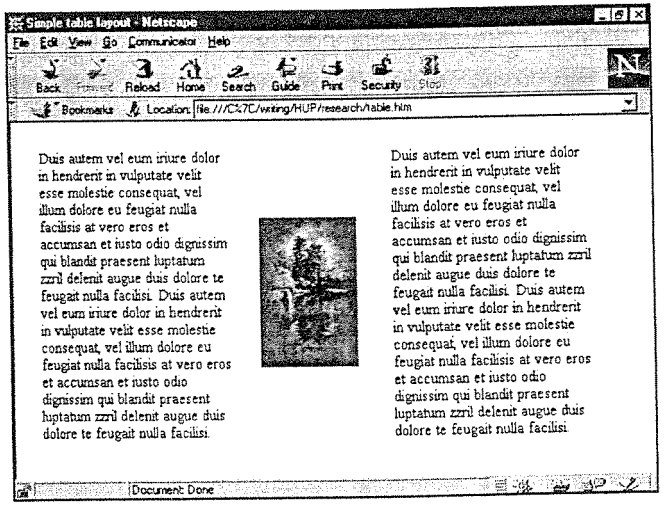
Folosirea tabelelelor pentru crearea unei machete simple

În această machetă sunt patru elemente: două imagini și două secțiuni de text. Doresc să realizez un aspect interesant, cu suficient spațiu alb între elemente. Pentru a realiza acest lucru, introduc o lățime adecvată pentru tabel precum și atributele `cellpadding` și `cellspacing`, care-mi oferă spațiu suplimentar în jurul elementelor. Am folosit de asemenea alinierea, bazându-mă pe eticheta `<ALIGN>` *în interiorul* celulei tabelului, în loc să folosesc limbajul standard HTML, fără tabel.

```
<html>
<head>
<title>A Simple Table Layout</title>
</head>

<body>

<table border=0 width=590 cellpadding=10 cellspacing=10>
<tr>

<td>
Duis autem vel eum iriure dolor in hendrerit in vulputate velit esse molestie
consequat, vel illum dolore eu feugiat nulla facilisis at vero eros et accumsan
et iusto odio dignissim qui blandit praesent luptatum zzril delenit augue duis
dolore te feugait nulla facilisi.
Duis autem vel eum iriure dolor in hendrerit in vulputate velit esse molestie
consequat, vel illum dolore eu feugiat nulla facilisis at vero eros et accumsan
et iusto odio dignissim qui blandit praesent luptatum zzril delenit augue duis
dolore te feugait nulla facilisi.
</td>

<img src="images/seattle.jpg" width=100 height=148 border=0 alt="seattle fall">

</td>

<td>

Duis autem vel eum iriure dolor in hendrerit in vulputate velit esse molestie
consequat, vel illum dolore eu feugiat nulla facilisis at vero eros et accumsan
et iusto odio dignissim qui blandit praesent luptatum zzril delenit augue duis
dolore te feugait nulla facilisi.
Duis autem vel eum iriure dolor in hendrerit in vulputate velit esse molestie
consequat, vel illum dolore eu feugiat nulla facilisis at vero eros et accumsan
et iusto odio dignissim qui blandit praesent luptatum zzril delenit augue duis
dolore te feugait nulla facilisi.

</td>

</tr>
</table>

</body>
</html>
```

După cum vedeți în Figura 34.16, rezultatele sunt sofisticate și este destul spațiu alb care să-mi satisfacă exigența.

FIGURA 34.16. *Folosirea tabelului pentru crearea unei machete atractive cu spațiu liber suficient.*



Folosirea cadrelor pentru crearea unei machete

Majoritatea persoanelor se gândesc la cadre cu chenar când este vorba despre cadre. În cazul de față, s-a folosit un procedeu fără chenar pentru a obține rezultatele prezentate în Figura 34.17.

FIGURA 34.17. *Cadrele fără chenar pot crea interfețe funcționale atractive.*

După cum se poate vedea, figura nu sugerează că aspectul, impresia și funcționalitatea au la bază cadrele. Din cauza folosirii cadrelor, porțiunea de navigare de pe partea stângă rămâne statică, iar paginile de informații se încarcă în partea dreaptă pe măsură ce vizitatorul selectează opțiunile de navigare.

Forma

Gândiți-vă un moment la semnificația cuvântului *formă*. Ați auzit cu toții spunându-se cam așa:

„E într-o formă excelentă."

„Ea are forme frumoase."

„În ce formă se prezintă *de fapt* această mașină uzată?"

Acest termen nu este folosit întâmplător în lumea noastră. El se referă la formele geometrice propriu-zise, la estetica aspectului, precum și la calitatea unui produs. Mai mult, forma are semnificații arhetipice – remanențe din trecut ale moștenirii culturale, care ne afectează foarte puternic, la nivelul subconștientului.

Studenții care studiază mijloacele de informare, publicitatea și designul sunt adesea informați despre puterea formei, dar dumneavoastră nu citiți prea des despre acest lucru în cărțile de proiectare pentru Web. Eu personal nu cred că există un loc mai puternic pentru a folosi efectiv forma, deoarece Webul însuși este un mediu cu potențial mare. Adăugați la aceasta o concepție inteligentă, bine gândită și situl dumneavoastră Web va avea un impact nebănuit.

Nu demult, când căutam să-mi cumpăr o mașină, am surfat prin diferite situri pentru mașini. Experiența a fost bogată, deoarece am văzut forma la lucru nu numai în campania de publicitate pentru mașini, dar și în proiectarea mașinilor!

După ce voi descrie sensul arhetipic al formei, voi demonstra eficacitatea utilizării formei în exemplele de pagini Web pe care le-am furnizat:

Cercul: Cercurile sugerează senzația de mișcare dar și leagătura istorică cu confortul, feminitatea, comunitatea și un tot unitar. Figura 34.18 prezintă concepția circulară a paginii Honda, care sugerează senzația de mișcare și libertate.

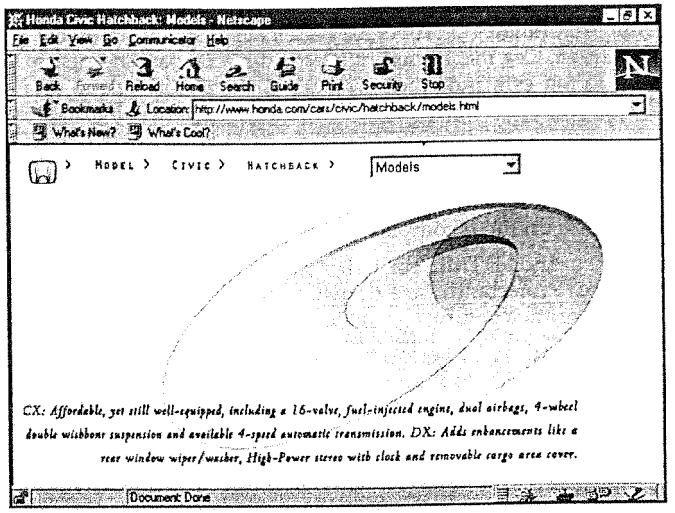


FIGURA 34.18. *Elementele circulare folosite în proiectare, la* `http://www.honda.com`.

Triunghiul: Triunghiurile se folosesc pentru atragerea atenției sau crearea senzației de mișcare în sus și sunt legate simbolistic de putere, știință și religie. La proiectarea mașinilor Saturn s-a folosit forma triunghiulară (vezi Figura 34.19). Este cumva de mirare faptul că Saturn are un grup de admiratori aproape religioși?

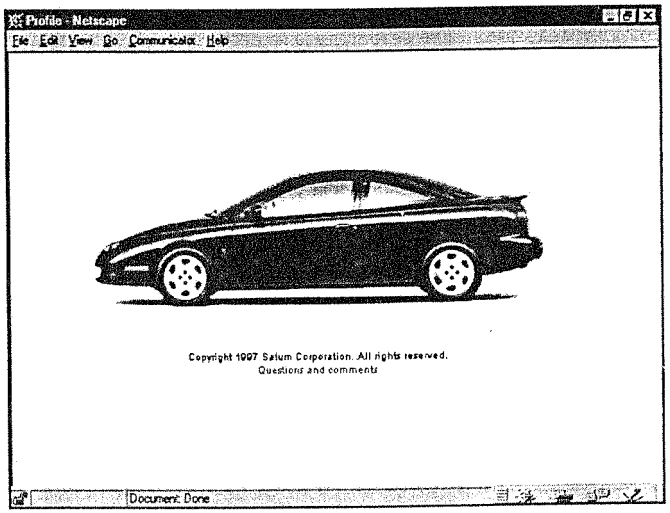


FIGURA 34.19. *Concepția autoturismului Saturn folosește din abundență triunghiurile, la* `http://www.saturncars.com/`.

Dreptunghiul: Semnificațiile principale ale dreptunghiului sunt stabilitatea și permanența. Marea problemă cu dreptunghiurile este că ele se află peste tot în Web!

Aceasta nu înseamnă că nu le puteți folosi, dar deprinderea modului de integrare a altor forme și combinații de forme poate fi în beneficiul dumneavoastră.

Așadar, cum obțineți forma pentru pagina dumneavoastră Web? În cea mai mare parte, forma va fi furnizată de către grafică. Totuși, spațiul alb și celelalte aspecte discutate anterior, precum proximitatea și proporția, joacă toate un rol în formele mai puțin evidente care apar într-o pagină. Cea mai importantă lecție din acest paragraf cu care trebuie să plecăm acasă este că forma este puternică. Trebuie să vă gândiți bine la ceea ce vreți să transmiteți audienței și apoi să folosiți forma ca instrument pentru îmbunătățirea concepției sitului Web.

Tipografia

Cunosc graficieni care au studiat tipurile de caractere de-a lungul întregii lor cariere. Fără glumă – este un lucru complex și o artă.

Până de curând, HTML era foarte limitat în ceea ce privește posibilitățile de lucru cu caracterele. Proiectanții care doreau să folosească stilurile tipografice se bazau în mare măsură pe grafică pentru a produce stiluri. Din fericire acest lucru a început să se schimbe.

În primul rând, există eticheta <FONT>, care implică multe discuții în ceea ce privește compatibilitatea la scară largă, dar permite proiectanților să abordeze tehnicile privind caracterele prin limbajul HTML. Eticheta <FONT> are mai multe atribute, printre care tipul caracterului, mărimea și culoarea.

Mai recent, au ajuns la modă foile de stil, care pot oferi fanaticilor fontului un control chiar și mai mare. În acest capitol, mă voi concentra mai întâi pe folosirea etichetei <FONT>, voi face unele referiri la foile de stil și vă voi recomanda o diversitate de surse unde puteți afla mai multe despre tipografia în Web și foile de stil. Pentru mai multe informații referitoare la utilizarea foilor de stil, consultați Partea a-IV-a a acestei cărți, „Tehnici de prezentare", care pune accentul pe tehnici de prezentare în HTML 4.0 mai îndrăznețe.

OBSERVAȚIE

Pentru a vedea toate tipurile de fonturi sau foi de stil vă trebuie, bineînțeles, un browser compatibil. Cei care folosesc Internet Explorer 3.0 sau Netscape 4.0 vor putea să vadă fonturile și foile de stil.

Tipul caracterelor

Un tip de caracter este un font: un model de caracter cu un anumit stil. Tipul caracterelor crește interesul vizual, identitatea și stilul unei pagini.

Tipurile de caractere sunt grupate pe mai multe categorii, numite „familii". În cadrul unei familii sunt mai multe tipuri de caractere, fiecare cu propria personalitate. Dacă doriți să studiați mai detaliat tipografia, veți găsi mai multe variații în cadrul unui anumit tip, acesta fiind unul dintre motivele pentru care studiul tipografiei este destul de serios!

Următoarele familii de tipuri vor prezenta importanță pentru dumneavoastră când veți proiecta un sit Web:

Serif: Aceste fonturi au liniuțe la extremitățile literelor. Un exemplu de font serif este cunoscutul Times Roman. Tipurile de caractere serif sunt considerate adecvate pentru citit, probabil datorită liniuțelor care ajută ochiul să le urmărească dar nimeni nu cunoaște în realitate de ce testele de citire au arătat că fonturile serif folosite ca fonturi obișnuite sunt superioare – și aceste teste au fost făcute *înainte* de apariția siturilor Web. Actualmente, veți găsi mult mai multe tipuri de font fără serif folosite în Web pentru corpul textului – ceea ce reprezintă o tendință interesantă.

Sans Serif: Această familie de fonturi nu are liniuțe la extremitățile literelor; de fapt, caracterele fără serife sunt rotunde și netede. Tipuri cunoscute de fonturi sans serif sunt Arial sau Helvetica. Aceste fonturi sunt ușoare pentru ochi și au fost folosite în mod tradițional mai mult pentru anteturi decât pentru corpul textului. Din nou, folosirea lor în mediul Web determină o schimbare în această tradiție.

Decorative: Fonturile decorative au caracteristici suplimentare, cum ar fi linii mai puțin obișnuite, influențe caligrafice, variații ale dimensiunilor și forme caracteristice. Tipul de caractere decorativ tinde să fie folosit cel mai bine după cum este chiar denumit: pentru decorații și sporirea complexității. Rar sau niciodată poate alcătui un bun corp de text, deoarece creează un aspect încărcat și aglomerat.

Script: Fonturile script seamănă cu scrisul cursiv. Reprezintă o alegere proastă pentru corpul textului, dar pot aduce îmbunătățiri substanțiale anteturilor sau zonelor în care scrisul e folosit ca element de creație.

Monospațiat: Fiecare caracter din cadrul acestei familii ocupă exact același spațiu în comparație cu celelalte. Litera *M* are de obicei o lățime diferită de *I*, dar în cazul fonturilor monospațiate lățimea e potrivită astfel încât să fie egală pentru fiecare caracter. Fonturile monospațiu au fost preferate în ultimii ani atât pentru corpul textului, cât și pentru antet.

Figura 34.20 prezintă câte un eșantion din fiecare tip de font.

FIGURA 34.20. *Eșantioane din diferite familii de fonturi.*

ATENȚIE!

Amestecarea tipurilor de font este amuzantă și poate duce la rezultate profesionale atractive. Totuși, mulți începători exagerează și nu înțeleg că adeseori „mai puțin" este „mai mult" când este vorba despre amestecarea fonturilor. Începătorii fără experiență tipografică ar trebui să evite amestecarea fonturilor din aceeași familie – nu amestecați Garamond și Times, de exemplu, deoarece ambele sunt fonturi serif. Fonturile din aceeași familie sunt adesea prea apropiate pentru a crea contrast – ceea ce dumneavoastră doriți – și prea diferite pentru a da aspectul de continuitate, care este necesar.

Folosirea atributului face

Dacă doriți să aplicați un tip de caracter unui text selectat, puteți face acest lucru folosind atributul face și apoi definind numele fontului. Pare destul de ușor, și așa și este. Problema este însă că, dacă fontul nu este instalat pe calculatorul vizitatorului, acesta nu va vedea fontul specificat! De aceea, e bine să rămâneți la fonturile obișnuite când selectați atributele etichetei <FONT>.

Iată un exemplu de etichetă <FONT> care include atributul face:

```
<font face="arial">This text will appear in the Arial face</font>
```

Majoritatea persoanelor cu un PC pe care rulează Windows vor vedea textul precedent cu fonturi Arial, dar utilizatorii de Macintosh nu au fontul Arial. Fontul sans serif obișnuit pentru Mac este Helvetica. Din fericire, eticheta <FONT> permite introducerea ambelor tipuri de fonturi în argument. Browserul va căuta primul font și, dacă nu-l găsește, se va deplasa la următorul nume de font:

```
<font face="arial, helvetica">This text will appear as Arial or Helvetica,
depending upon which font is available</font>
```

Acum ați furnizat fonturile necesare celor care au PC, precum și celor care au Macintosh. Acesta este lucrul care trebuia făcut.

ATENȚIE!

Amintiți-vă că dacă un tip de font nu este disponibil pe calculator, va apărea tipul de font prestabilit. Tipul prestabilit este aproape întotdeauna un font serif, cum este Times, dacă utilizatorul nu a selectat un alt tip de font. Așa că, dacă amestecați fonturile, țineți minte că un font sans serif poate apărea ca serif și invers. Această lipsă de control poate părea o nebunie! Puteți să vă abțineți să folosiți un anume tip, dar sunteți în pericol de a fi plictisitor. Principala grijă pe care ar trebui să o aveți este să vă cunoașteți publicul, așa cum am mai spus; acest lucru vă va ajuta să faceți cele mai bune opțiuni în toate cazurile.

O dezvoltare interesantă în tipografia Web o reprezintă fonturile înglobate. Acest procedeu descarcă automat caracterele tipului de font necesare pentru a vizualiza o anumită pagină. În vara anului 1997 Microsoft a lansat programul preliminar Web Embedding Font Tool. Acest program analizează tipurile de fonturi dintr-un anumit sit, comprimă caracterele corecte ale fontului și furnizează autorului codul pentru a înglobe fonturile comprimate ca obiecte în codul HTML. Pare confuz? Este, puțin – dar rezultatul este un control mai bun asupra fonturilor. Pentru informații suplimentare, consultați http>//www.microsoft.com/typography/. Acest sit oferă multe informații valoroase privitoare la tipografia Web în general.

Mărimea caracterelor

Alt aspect de care trebuie ținut seama când se lucrează cu fonturi este mărimea lor, care are importanță mare pentru stil și concepție. Mărimea fontului poate ajuta în indicarea rolului jucat în pagină: fonturile mai mari sunt folosite pentru anteturi, cele de mărime medie pentru corpul textului, iar fonturile mici pentru observații sau informațiile mai puțin importante, cum sunt de exemplu notele privind drepturile de autor.

Variația mărimii tipului de caractere într-o pagină este importantă, dar tot atât de importantă este păstrarea consecvenței acestei variații. Dumneavoastră căutați să obțineți interesul vizual și rezultate practice prin diferite mărimi de fonturi într-o pagină; totuși, nu trebuie să copleșiți vizitatorul dumneavoastră cu prea mult contrast, care poate duce la un haos deplin!

Folosirea atributului size

Dimensionarea fonturilor e destul de rudimentară, mărimea fontului fiind determinată de valori ale numerelor întregi. Mărimea prestabilită, standard este 3. Orice număr mai mare va genera un font mai mare și orice număr mai mic va determina un font mai mic. Puteți folosi și numere negative, ca de exemplu –1, pentru a avea tipuri de caractere foarte mici. Iată un exemplu de antet care folosește atribute pentru tipul și mărimea fonturilor:

```
<font face="times,garamond" size=5>
```

Orice font mai mare de 5 este stângaci. Fonturile mici, cum este 1, sunt bune pentru note și drepturi de autor. Orice este mai mic nu poate fi citit de persoanele cu vedere între medie și slabă.

OBSERVAȚIE

Foile de stil în cascadă rezolvă limitările mărimii fontului, oferind posibilitatea de a folosi punctele și pixelii (ca și inchii sau centimetrii) pentru a măsura un font. Capitolul 18, „Utilizarea foilor de stil în cascadă" și 19, „Foile de stil în cascadă și poziționarea elementelor", din această carte pun accentul pe folosirea foilor de stil pentru prezentările profesionale.

Figura 34.21 prezintă un exemplu de antet, corp de text și notă pentru drepturi de autor, fiecare folosind o mărime diferită de font. Observați de asemenea și tipul fontului. Antetul este cu Times, iar nota pentru drepturile de autor este cu Times setat pe italic. Fontul pentru corpul textului este Arial. Această pagină arată destul de elegant, spre deosebire de Figura 34.22, care prezintă ce se întâmplă când programatorul o ia razna. Efectul „undă" a fost la modă odată, când a apărut posibilitatea de setare a mărimii fonturilor. Deși era distractiv la acea vreme, a devenit repede stereotip. Nu este nici prea inteligent din punct de vedere al concepției; nimeni nu va sta prea mult prin preajmă dacă trebuie să citească o pagină ca aceasta!

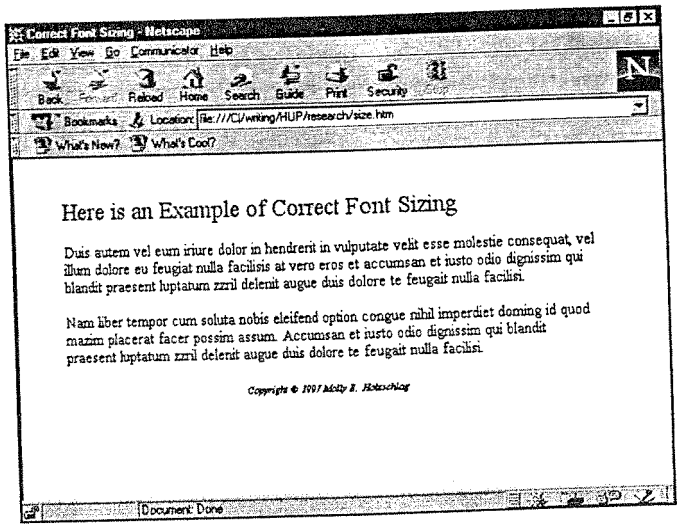
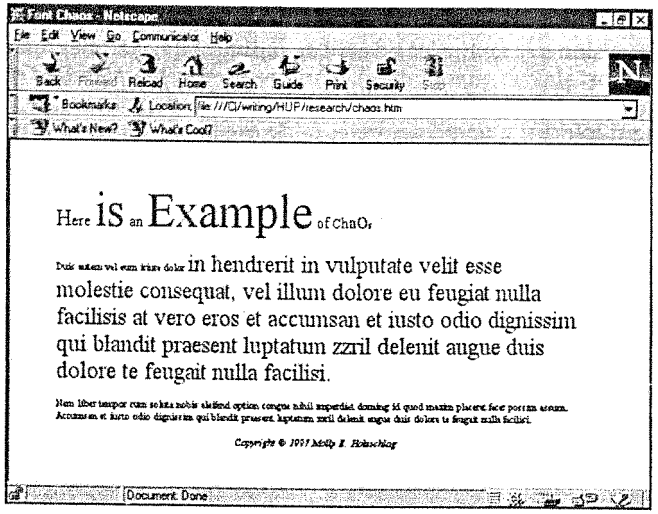


FIGURA 34.21. *Folosirea dimensionării fonturilor în mod atractiv și eficient.*

FIGURA 34.22.
Haosul fonturilor!



Culoarea fonturilor

Prin colorarea fonturilor puteți conferi distincție paginii. Ca și în cazul mărimii și tipului caracterelor, aceasta trebuie folosită cu măsură; nu trebuie să copleșiți vizitatorii sitului dumneavoastră cu 10 culori diferite pe aceeași pagină. De fapt, dacă folosiți numai două culori – una pentru anteturi și textul auxiliar și una pentru corpul textului – este suficient. Amintiți-vă că mai folosiți culori pentru legături și pentru legăturile vizitate, care sunt definite în cadrul etichetei `<BODY>`, așa că dacă folosiți mai mult de două culori pentru text, depășiți limitele normale.

Un exemplu al etichetei `<FONT>` conținând atributul de culoare arată astfel:

```
<font face="Times,Garamond" size=5 color="#003300">
```

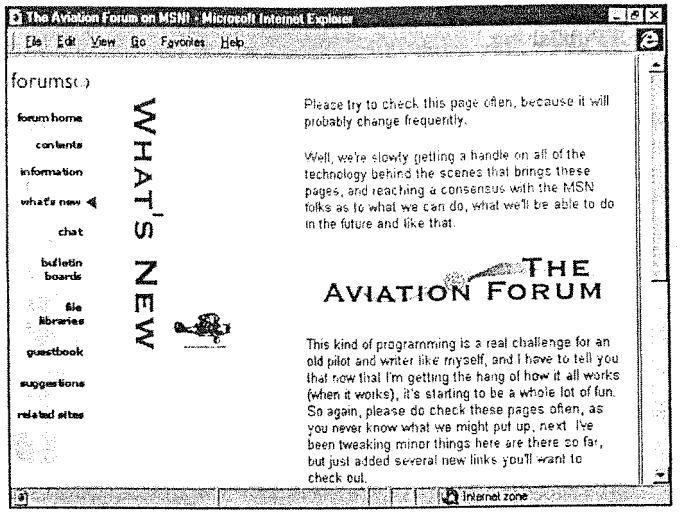
Folosind codul hexazecimal, echivalentul în baza 16 a valorilor RGB (red, green, blue), am selectat o nuanță de verde pentru antet. Puteți folosi denumirea culorilor standard, precum blue, green, red ș.a. Nu sunt adeptul acestei metode, deoarece e foarte limitată. Pentru o bună hartă de culori hexazecimală, vizitați `http>//:sdc.htrigg.smu.edu/HTMLPages/RGBchart.html` sau descărcați fișierul `nvalue.gif` sau `nhue.gif` de la `http://www.lynda.com/files/`. Aceste hărți vă pun la dispoziție selecția culorilor și valorile hexazecimale.

Alte considerații tipografice

Trebuie luate în considerare și alte elemente când este vorba despre tipografie. Vestea bună este că ele sunt avute în vedere de tehnologii noi sau pe cale de apariție. Vestea proastă este că unele dintre ele pot fi realizate numai folosind grafica. Iată pe scurt unele dintre aceste considerații:

- Modificarea direcției textului poate produce un efect interesant. Figura 34.23 prezintă un font care se afișează pe verticală. Acest efect a fost realizat prin folosirea procedeelor grafice, pentru că limbajul standard HTML nu îl permite.

FIGURA 34.23.
Antetul vertical pe această pagină a fost realizat printr-un procedeu grafic.



- Tipografia se ocupă, de asemenea, cu spațiul dintre linii, denumit *interliniere* (leading). Interlinierea poate fi stabilită cu ajutorul unui atribut al foii de stil denumit *line height (înălțimea liniei)*. În limbajul HTML standard nu există o opțiune corespunzătoare pentru interliniere.
- Modificarea spațiului dintre litere (kerning). Cu excepția folosirii procedeelor grafice, nu există soluție pentru acest lucru.
- Altă problemă este dezimțuirea (anti-aliasing), cunoscută în lumea calculatoarelor și sub numele de *netezirea fonturilor*. Fără dezimțuire, caracterele pot avea un aspect neșlefuit. Figura 34.24 prezintă un antet nongrafic – puteți vedea muchiile neșlefuite ale caracterelor. Aș putea folosi o metodă grafică și să selectez opțiunea pentru dezimțuire când pregătesc textul pentru grafică. Aceasta este o soluție bună. Altă soluție este controlul zimțuirii, cum a făcut de exemplu Microsoft cu programul de netezire a fonturilor, care poate fi descărcat pentru calculatoare PC cu Windows 95 de la `http://www.microsoft.com/typography/grayscal/smoother.htm`. Figura 34.25 prezintă același antet, vizualizat cu opțiunea de netezire a caracterelor selectată.

Dacă vă interesează într-adevăr tipografia pe Web, iată câteva resurse pentru a începe:

Pe Internet

DesktopPublishing.Com: un loc într-adevăr minunat, la `http://www.desktoppublishing.com/`.

World Wide Web Consortium: Tot felul de resurse HTML, la `http://www.w3.org`. Pentru o privire de ansamblu asupra foilor de stil în cascadă, vizitați `http://www.w3.org/style/`.

Tipografia Microsoft pe Web: O resursă excelentă pentru tot ce se întâmplă în tipografia Web, la `http://www.microsoft.com/typography/web/default.htm`.

Tipografia Web: Un rezumat acceptabil, la `http://acdcon.com/webtyp.htm`.

Cărți

The Mac is not a Typewriter(de asemenea *The PC is not a Typewriter)*, Robin Williams, Peachpit Press.

Desktop Publisher's Easy Type Guide, Don Dewsnap, Rockport Publishing.

Alphabet: The History, Evolution, and Design of the Letters We Use Today, Allan Haley, Watson Guptill Publishing.

Type and Layout, Colin Wheildon, Strathmore Press.

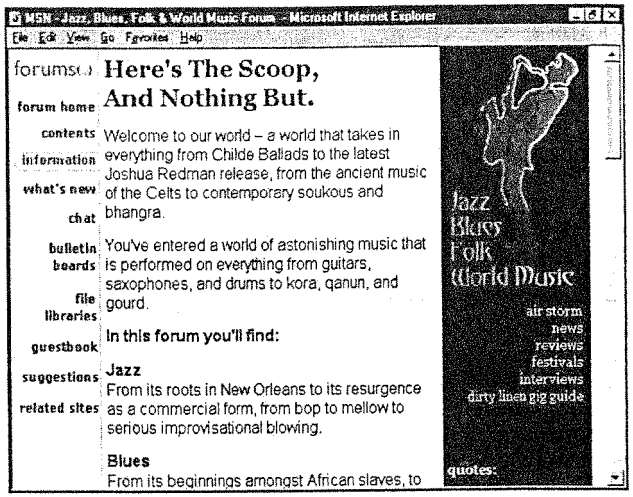


FIGURA 34.24. *Antetul paginii fără opțiunea de netezire a fonturilor.*

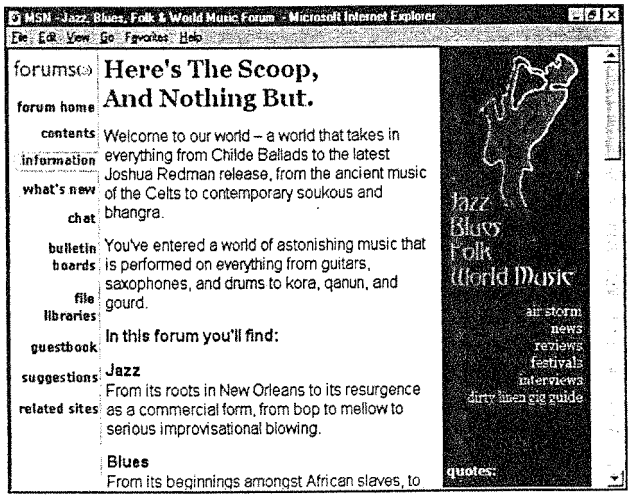


FIGURA 34.25. *Același antet cu opțiunea de netezire a fonturilor selectată.*

Rezumat

Păstrați în formă paginile dumneavoastră, acordați-le ceva spațiu și adăugați mult stil prin procedee corespunzătoare de proiectare! Cele mai multe procedee din acest capitol apelează la metode proprii limbajului HTML standard, pe care le puteți folosi în acest moment. Altele, cum sunt foile de stil în cascadă, necesită mai multă cercetare, dar în orice caz, sunteți pe drumul care vă poartă către concepția unui sit Web mai aspectuos și mai eficient.

Capitolul 35

Toate puse împreună: forumul DisAbilities din Microsoft Network

de Molly E. Holzschlag

În acest capitol:

- **Prezentarea interfeței cu forumul DisAbilities** 539
- **Aspectele generale ale planificării sitului** 544
- **Elemente interactive și multimedia** 547
- **Navigarea** 549
- **Lucrul cu culorile și grafica** 556
- **Spațiul, forma și fonturile** 558

Acum, după ce ați învățat o gamă de procedee care vă ajută la îmbunătățirea sitului dumneavoastră Web, veți examina în detaliu un sit Web pentru a vedea aceste procedee la lucru.

Am ales forumul DisAbilities din Microsoft Network (MSN) deoarece este unul din cele mai mari situri pe care le-am realizat, cu mai multe secțiuni mai mici. Folosește de asemenea o gamă largă de medii și programare, inclusiv RealAudio, VBScript, Macromedia Flash și multă interactivitate, cum ar fi camerele de discuții în direct și avizierele electronice. Doresc să vă arăt cum se păstrează continuitatea în fiecare secțiune, dar și individualitatea, și să demonstrez cum își menține situl integrarea prin interfață, navigare și elementele de proiectare, în ciuda componentelor sale interactive.

Prezentarea interfeței cu forumul DisAbilities

După cum am menționat în Capitolul 31, „Realizarea unei interfețe eficiente", utilizatorii trebuie să fie întâmpinați, conduși și să li se ofere elemente funcționale pentru a se conecta cu succes la sit prin intermediul interfeței.

Figura 35.1 prezintă pagina de întâmpinare a forumului DisAbilities. Această pagină funcțională stabilește interfața, introducând imediat un număr de opțiuni care poartă vizitatorul direct la zone care ar fi de interes pentru el. Această pagină se schimbă periodic pentru a reflecta gama de activități oferite în sit.

În acest caz, nu numai că pe marginea din stânga s-a introdus navigarea prin cadre, dar și secțiunile acestei pagini principale statice (fără defilare) acționează ca pasaje de trecere către alte zone ale sitului. Desenul cu soarele zâmbitor, din partea stângă-sus a paginii de întâmpinare, face legătura cu festivalul de vară, un eveniment promoțional crucial care face legătura cu alte activități din rețea. Sub acesta, este o legătură la propria activitate de vară a forumului DisAbilities, festivalul Creative Arts, pe care-l voi studia mai îndeaproape în acest capitol.

În dreapta-sus este o animație de întâmpinare, care spune vizitatorului unde se află și face o declarație privitoare la ceea ce trebuie să se aștepte de la conținutul sitului, prin sloganul principal al sitului: „No Fear, No Pity, No Limits" (fără teamă, fără milă, fără limite).

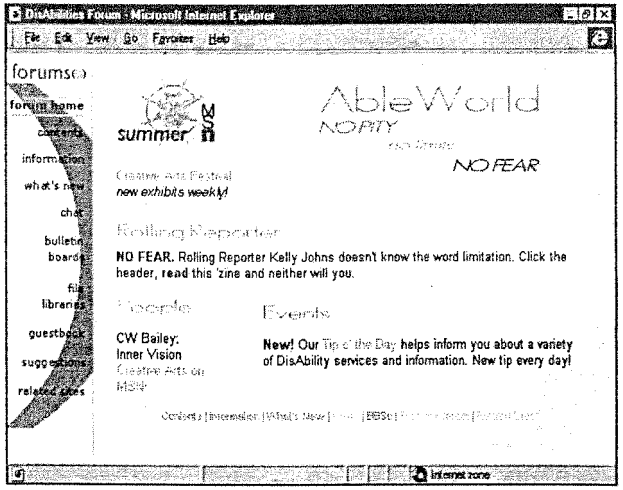


FIGURA 35.1. *Pagina de întâmpinare a forumului DisAbilities.*

O altă componentă importantă a forumului este Rolling Reporter, o revistă de călătorie și aventuri, destinată persoanelor cu handicap. Locul său proeminent pe această pagină atrage vizitatorul, făcându-i cunoscut că este un loc de vizitat important! Veți vedea Rolling Reporter mai îndeaproape pe măsură ce turul cu îndrumător continuă.

În partea de jos, două „uși" oferă alte activități. Una este secțiunea Events (evenimente), care se schimbă, de asemenea, periodic. În acest caz, oferă o legătură la sfatul zilei din forumul DisAbilities, o pagină zilnică bazată pe cadre care prezintă un sfat, după cum se vede în Figura 35.2.

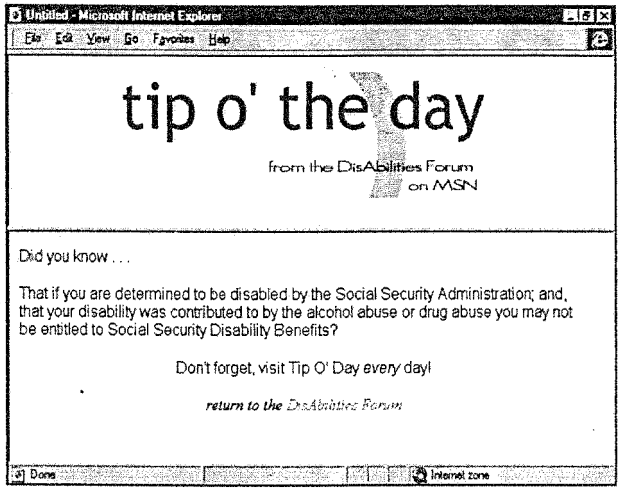


FIGURA 35.2. *Pagina cu sfatul zilei.*

Altă ușă conturată pe prima pagină este secțiunea People (personalități), care are o legătură dinamică către informații privind diferiți membri și personal din forum, ca și oaspeți de vază ai forumului. Acest exemplu conține legătura cu activitatea fotografului

C.W. Bailey, prezentat în festivalul de artă. Executați clic pe acea legătură și iată! Sunteți un vizitator al expoziției lui C.W., prezentată în Figura 35.3.

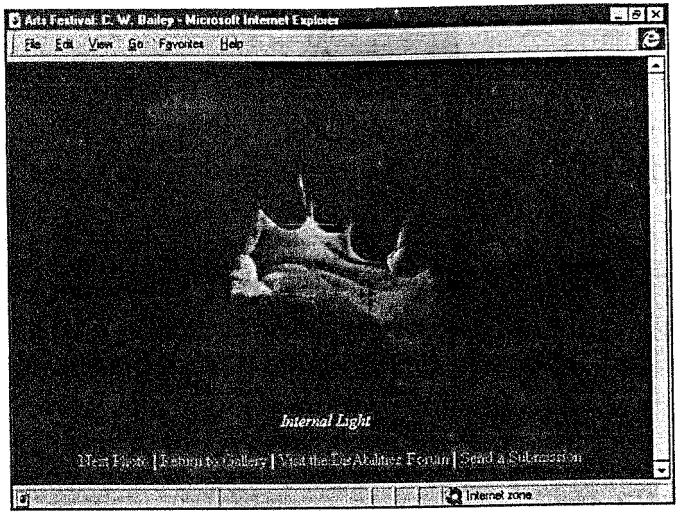


FIGURA 35.3. *Legătura la expoziția fotografică a lui C.W. Bailey, din secțiunea People.*

În concluzie, interfața cu pagina principală întâmpină vizitatorul, îl conduce către locurile importante și cunoscute din sit prin bara de navigare din partea stângă și scoate în evidență zonele prin „uși" virtuale pe prima pagină. Această pagină de întâmpinare prezintă vizitatorilor elementele funcționale ale sitului, dar și pe cele estetice.

Metafora

Metafora de proiectare folosită în planul principal al forumului DisAbilities este foarte subtilă. Nu s-au folosit pictogramele din două motive: primul, navigarea a fost predeterminată de rețea și, în al doilea rând, pentru multe persoane cu handicap explicațiile prin text sunt mai eficace decât indiciile vizuale. Sunt totuși locuri în care folosirea metaforei este foarte avantajoasă.

Interfața cu utilizatorul a sitului revistei Rolling Reporter folosește în mod clar metafora în proiectare. Priviți simbolul inteligent din figura 35.4. Având la bază simbolul internațional pentru persoanele cu handicap, s-au făcut puține schimbări pentru a folosi metafora în simbol. Roata scaunului este Pământul, iar silueta din scaun este apleacată înainte în timp ce scaunul se deplasează prin spațiu. Această metaforă sugerează faptul că, într-un fel, lumea este accesibilă tuturor, iar problematica abordată de sit este de viitor și progresistă. În final, stelele care clipesc liniștit în fundalul din spatele siluetei care plutește sugerează faptul că pot fi atinse – o metaforă într-adevăr pozitivă și puternică.

Claritatea și consecvența

Claritatea este un aspect critic al concepției interfeței, așa că, în forumul DisAbilities, butoanele, hărțile și legăturile sunt preponderent sub formă de text și descrise în mod explicit. De exemplu, dacă doriți informații privitoare la forum, executați clic pe `information` în bara de navigare de pe marginea din stânga. Această acțiune oferă opțiuni de poștă care derulează un cititor de poștă, după cum se arată în Figura 35.5. Textul este concis, descriptiv și intenționat menținut scurt, pentru a păstra claritatea.

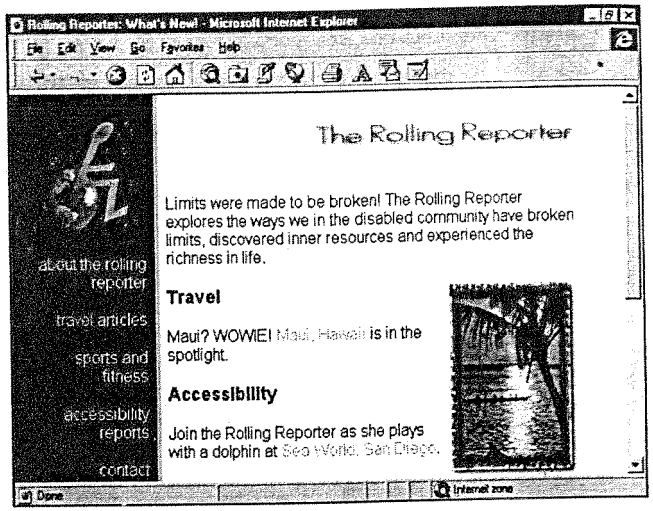


FIGURA 35.4. *Revista Rolling Reporter.*

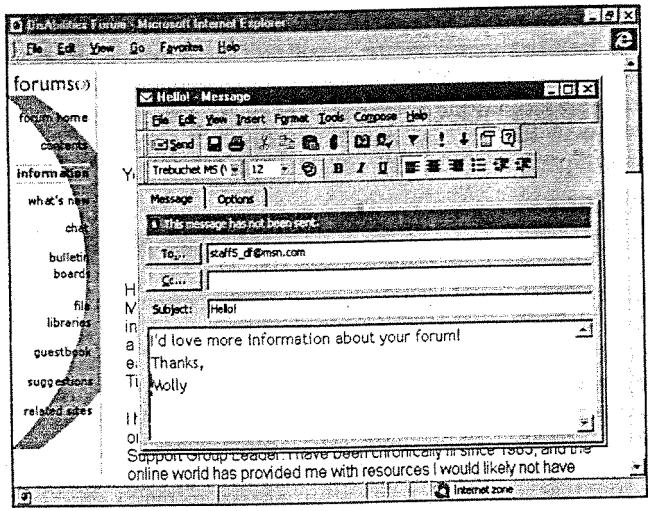


FIGURA 35.5. *O opțiune pentru poștă derulează un cititor de poștă.*

Elementele de proiectare ajută la realizarea consecvenței. La nivelul principal al forumului, toate anteturile grafice, corpul textului și fundalul rămân neschimbate. Comparați Figurile 35.6 și 35.7. Deși reprezintă pagini diferite, cu conținut diferit, elementele de concepție sunt consecvente, ceea ce dă sens integrării. Observați, de asemenea, pe ambele pagini, folosirea părții din stânga a ecranului pentru navigare.

Consecvența între secțiunile sitului e menținută prin folosirea aceluiași tip de font și a aceleiași culori pentru majoritatea zonelor forumului. Festivalul Creative Arts face excepție, deoarece, pentru a sugera creativitatea, unele elemente trebuie eliberate. Totuși, deși elemente ca fundalul și fonturile diferă de restul sitului, consecvența e păstrată prin menținerea paletelor de culoare ale sitului.

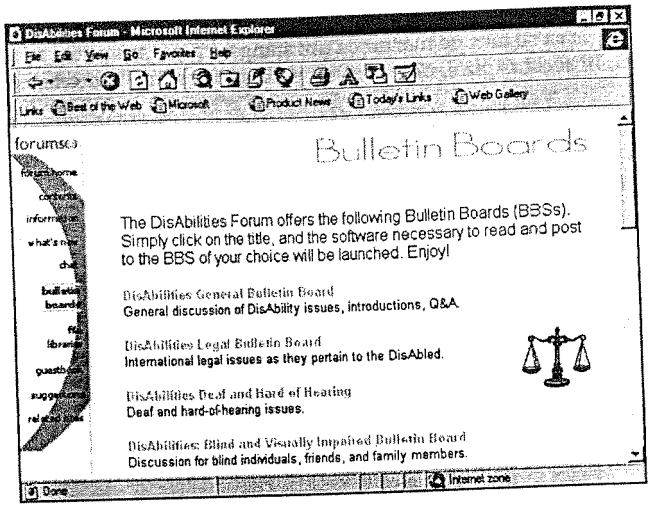


FIGURA 35.6. *Pagina BBS (sistem de avizier electronic).*

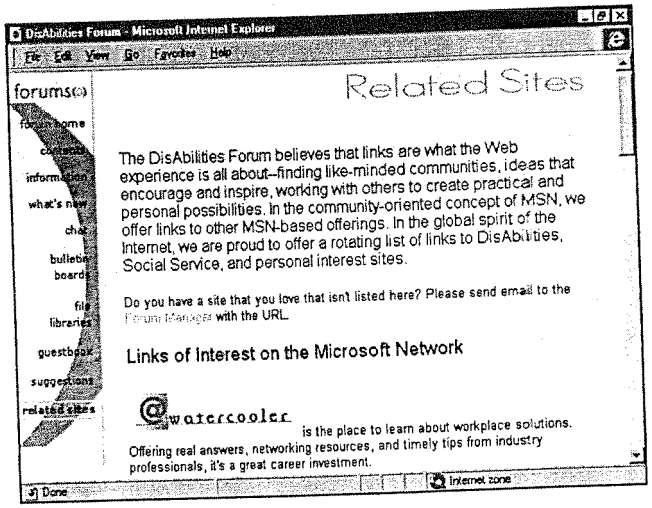


FIGURA 35.7. *Pagina cu situri înrudite.*

Orientarea și navigarea

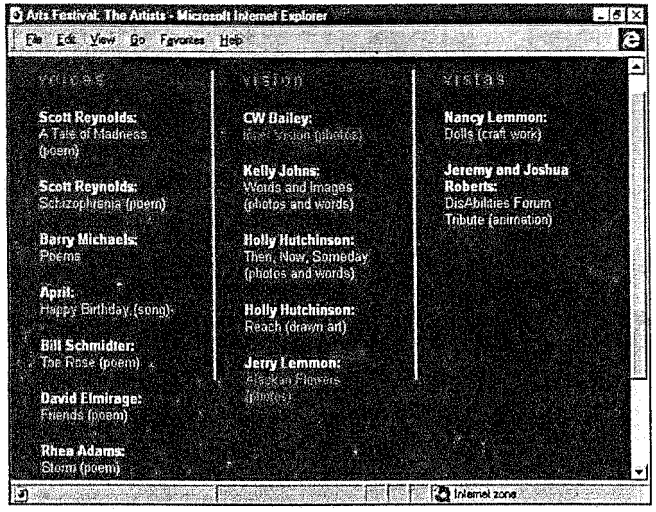
Orientarea este realizată la nivelul superior al acestui sit, prin bara de navigare omniprezentă, care rămâne statică în tot situl. După cum vedeți în figura 35.7, navigarea la nivelul superior are și o componentă text în partea de jos a ecranului, care ajută la orientare prin întărirea secțiunilor sitului.

Bara de navigare principală a fost creată de Microsoft Network folosind produsul Macromedia's Flash (`http://www.macromedia.com/`). Acest produs a fost folosit și pentru a crea butoanele de pe prima pagină. Acest program de animație compact și ușor de folosit realizează programe pentru diferite medii, iar vestea bună este că browserele Netscape și Internet Explorer (varianta 3.0 sau cele ulterioare) pot lucra cu aceste fișiere care se încarcă rapid.

În rest, navigarea prin forum este bazată pe text. Dacă vă uitați la Figura 35.4, veți observa că navigarea se face pe marginea din stânga. Vă voi arăta cum a fost făcut acest lucru, mai târziu, în acest capitol, când voi lua în discuție controlul machetării cu tabele și cadre.

Festivalul Creative Arts se bazează pe meniul principal (vezi Figura 35.8), care servește ca un index pentru toate paginile din situl festivalului. Navigarea în situl festivalului e bazată pe text, cu opțiuni adiționale asupra prezentărilor individuale și oferind intotdeauna legături înapoi la index sau la nivelul superior al forumului.

FIGURA 35.8.
Meniul principal al festivalului Creative Arts.



În sfârșit, vizitatorii se pot orienta vizual singuri datorită consecvenței paletei și a stilului, precum și datorită barei de titlu și identificării pe bază de text care e derulată în partea de jos a paginii (vezi Figura 35.9).

Aspectele generale ale planificării sitului

Acum, că avem o bună imagine asupra interfeței și a componentelor de bază ale concepției forumului, voi face un pas înapoi pentru un moment, pentru a arunca o privire asupra modului de planificare a sitului. Țineți minte, numai planificarea atentă va asigura aspectul și senzația de consecvență și integritate, pe care le căutați în proiectare.

Scopul proiectului

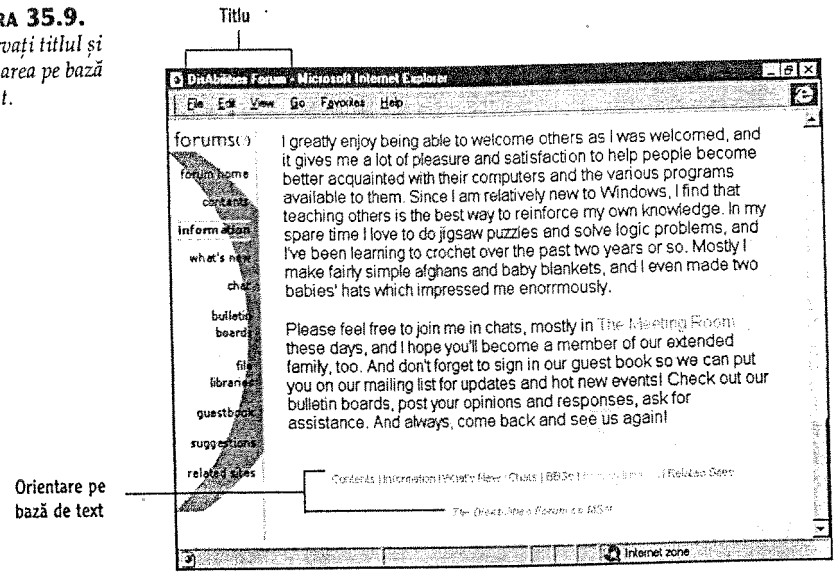
Scopul forumului DisAbilities este să informeze, să educe și să acorde sprijin persoanelor cu handicap sau celor a căror viață este într-un fel afectată de un handicap. Acest scop este abordat printr-un număr de opțiuni și servicii care formează *conținutul* forumului.

Conținutul proiectului

La cel mai de sus nivel, conținutul e în primul rând informațional, având o interfață care deschide căi de acces către activitățile și resursele găsite în sit. Informațiile de pe nivelul cel mai de sus, listate în bara de navigare, cuprind următoarele:

Contents (Cuprins): Acesta este un index al opțiunilor sitului.

FIGURA 35.9.
Observați titlul și orientarea pe bază de text.



Information (Informații): Zona de informare prezintă personalul forumului și furnizează informații asupra modului de folosire a forumului.

What's new (Ce e nou): Această secțiune ajută vizitatorii să găsească repede ce e nou și interesant în forumul DisAbilities.

Chat (Discuții): În această pagină este disponibilă o listă a temelor de discuții, una a camerelor de discuții și informații generale privind discuțiile.

Bulletin Boards (Aviziere electronice): Sunt disponibile descrieri individuale ale BBS (sisteme de avizier electronic) și legături către fiecare.

File libraries (Biblioteci de fișiere): Din această pagină poate fi accesată fiecare bibliotecă de fișiere. Fișierele din situl forumului cuprind documente cu informații privitoare la diferite forme de handicap, legi, beneficii și programe software speciale.

Guest Book (Cartea de oaspeți): Această zonă este destinată prezentării membrilor.

Suggestions (Sugestii): Această zonă este cea mai importantă secțiune de reacții din forum. Aici își pot depune vizitatorii ideile sau preocupările privind forumul.

Related sites (Situri înrudite): Această zonă oferă legături către alte locuri din Microsoft Network și alte situri Web specializate, de interes pentru comunitate.

OBSERVAȚIE

Acest nivel înalt de prezentare reprezintă un bun exemplu pe care vă puteți baza când construiți situl dumneavoastră. Nu oricine va avea BBS (sistem de avizier electronic) sau domenii de fișiere, dar structura generală a acestui sit este adaptată proiectelor Web obișnuite.

Conținutul cu caracter special este furnizat de caracteristicile paginilor festivalului de artă, ale revistei Rolling Reporter și ale sfatului zilei.

Scopurile pe termen scurt și lung

Scopul pe termen scurt a fost simplu: să furnizeze un punct de plecare către diferitele aspecte ale conținutului forumului astfel încât membrii să obțină rapid informația de care au nevoie.

Scopul pe termen lung se constituie pe baza semnificației sitului. Conținutul este compus prin crearea – și păstrarea – rezultatelor activităților și evenimentelor. De exemplu, festivalul Creative Arts a început ca o parte a unui eveniment sezonier (în acest caz, estival), dar conținutul interesant, colectat pentru acest eveniment rămâne disponibil ca o expoziție care continuă. În mod similar, toate articolele și legăturile adăugate sitului Rolling Reporter sunt realizate astfel încât părțile interesate pot să le găsească ușor.

Colectarea și arhivarea conținutului creează o resursă în creștere, vitală pentru comunitatea persoanelor cu handicap din Microsoft Network. Gândiți-vă la scopurile sitului dumneavoastră. Dumneavoastră doriți ca persoanele să se întoarcă la situl dumneavoastră și să-l utilizeze ca punct de plecare, referință, distracție sau ca mecanism pentru vânzarea produselor. Oferind informații importante publicului dumneavoastră, măriți eficacitatea sitului dumneavoastră Web. În cazul forumului DisAbilities, membrii vor ști că dacă, de exemplu, intenționează să facă o călătorie la Disney World, este probabil să existe un articol în Rolling Reporter care-i va ajuta să ia decizii privitoare la călătorie (vezi Figura 35.10).

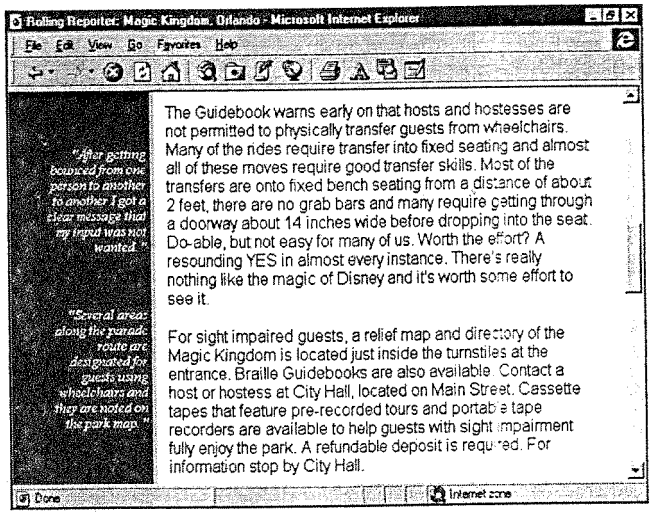


FIGURA 35.10. *Doriți să aflați date referitoare la o vizită accesibilă la Disney World? Rolling Reporter este locul.*

Experiența publicului

Forumul nu este conceput numai să fie plin de resurse, dar și să fie un loc unde comunitatea să se adune, să se ajute unul pe altul în momentele de dificultate și să învețe unul de la altul procedee și metode eficiente pentru viața de zi cu zi.

Toate aceste activități dau senzația de *comunitate* și cum Webul devine din ce în ce mai mare, devin imperative promovarea și păstrarea acestei senzații de comunitate. Oamenii intră pe Internet sau apelează serviciile online nu doar pentru a căuta informații – ei doresc să interacționeze cu alții! Interacțiunea este cea mai mare fascinație din Web și dumneavoastră puteți aduna oamenii în jurul intereselor comune. Această fascinație și nevoia de

comunitate vă pot ajuta în căutarea succesului cu situl dumneavoastră Web. Gândiți-vă la comunitate și la experiența audienței când planificați și proiectați situl dumneavoastră.

Tema principală a forumului DisAbilities este promovarea cunoașterii. Publicul ar trebui să aibă senzația de sprijin colectiv ca și de acces la informație – în mod ideal o experiență mobilizatoare și benefică.

ATENȚIE!

Similar cu oricare aspect al concepției sitului Web, cunoașterea publicului dumneavoastră este critică. Dacă nu ați făcut analiza preliminară și nu ați stabilit *cine* vizitează situl dumneavoastră, va fi dificil să oferiți senzația de comunitate. Trebuie mai întâi să vă cunoașteți audiența și apoi să le oferiți ceea ce doresc și au nevoie.

Elemente interactive și multimedia

Resursele pline de interes, destinate comunităților sunt create în Web cu elemente interactive și evenimente multimedia. Trebuie să captați atenția publicului și să o mențineți! Forumul DisAbilities realizează acest lucru cu o gamă largă de evenimente interactive și multimedia, care vor fi prezentate în următoarele secțiuni.

Discuții și aviziere electronice

Microsoft Network este proprietarul pachetului software pentru crearea și utilizarea camerelor de discuții și a avizierelor electronice. În cazul forumului DisAbilities, aceste zone constituie adevărata inimă a comunității – ele fac oamenii să revină. (Vezi Figura 35.11). Se pot schimba informații și se poate acorda ajutor reciproc, dar, ce este mai important, se dezvoltă relații de prietenie.

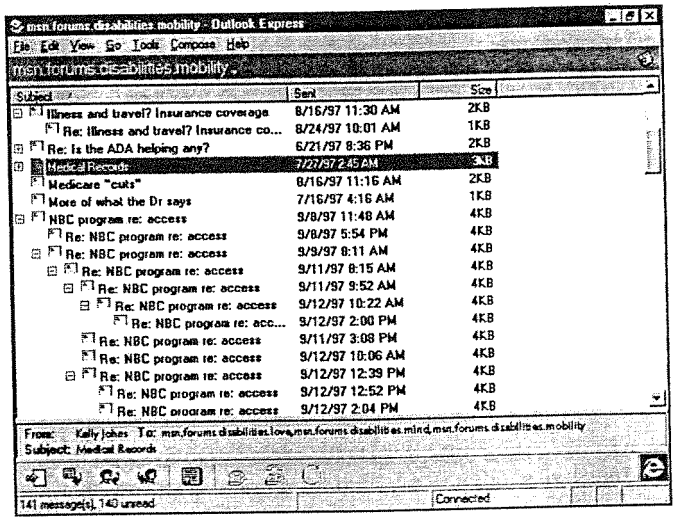


FIGURA 35.11. *Un sistem de avizier electronic din forumul DisAbilities.*

Mulți dintre membri forumului se întâlnesc personal și experiența lor de viață a fost îmbogățită de componentele online ale relației lor. În acest fel, un sit Web poate fi nu numai o afacere bună, dar și ajută la refacerea comunității folosind căi importante și durabile.

Probabil vă gândiți că acest lucru nu are ce căuta în situl dumneavoastră Web folosit ca punct de vânzare. V-aș atenționa să gândiți mai creativ, cum o fac multe situri comerciale, ca de exemplu situl Saturn Cars (`http://www.saturncars.com/`) sau Travelocity (`http://www.travelocity.com/`).

Acestea sunt două situri Web cu scopul principal de a vinde un produs – mașini sau servicii de călătorie – dar fiecare sprijină comunitatea prin furnizarea sistemului de avizier electronic pentru vizitatori. Am intrat recent într-o discuție online referitoare la camerele de hotel; oamenii își împărtășeau impresiile despre un anumit hotel în care urma să stau. Când am ajuns acolo, eram înarmată cu informații privitoare la hotel, știam cum să-mi optimizez timpul petrecut acolo, prin obținerea celei mai bune camere disponibile și cunoscând cui să mă adresez dacă aveam nevoie de ceva, precum și alte aspecte.

Pe scurt, fiți sigur că pe viitor mă voi reîntoarce în acest sit Web. Știu cât de mult mi-a îmbogățit experiența această interacțiune. O soluție inteligentă de creștere a vânzărilor!

Multimedia și programarea

O altă cale de a antrena publicul este oferta de elemente interesante și amuzante, programate în sit. În cazul forumului DisAbilities, multimedia e folosită cu grijă, deoarece publicul este caracteristic. De exemplu, deși Macromedia Flash și RealAudio (vezi Figura 35.12) sunt folosite în unele secțiuni ale sitului, evenimentele audio și video sunt menținute la un nivel minim, deoarece mulți membri ai forumului sunt orbi sau surzi. Totuși aceste opțiuni îmbunătățesc experiența celor care le pot accesa și se are grijă să nu fie alterat în mod semnificativ conținutul sitului din cauza existenței facilităților multimedia.

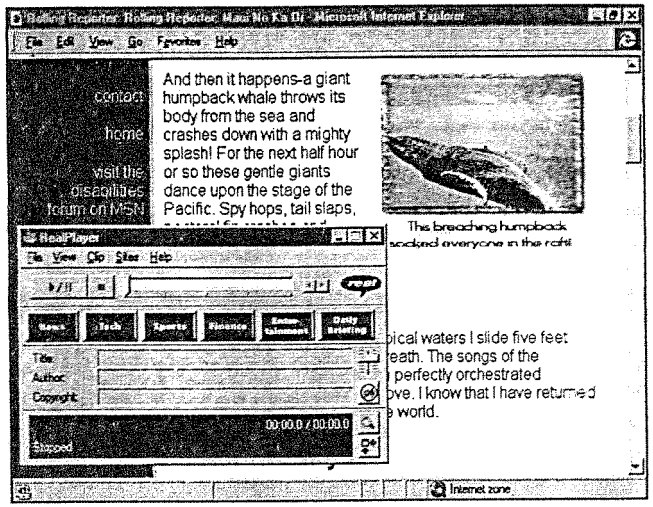


FIGURA 35.12. *Fișierul RealAudio care redă sunetul înregistrat când o balenă sare deasupra apei îmbunătățește experiența interactivă a revistei.*

Evenimentele create prin programare sporesc de asemenea interesul pentru sit. Tip o' the Day (Sfatul zilei) folosește VBScript pentru a livra informații importante. În fiecare zi, aceste informații se actualizează automat, ceea ce menține prospețimea sitului. Personalul este solicitat foarte puțin de întreținerea acestei caracteristici, iar publicului îi face plăcere să intre în sit și să găsească lucruri noi dar și utile.

Gândiți-vă la evenimentele multimedia și programate care vor îmbunătăți situl dumneavoastră și experiența vizitatorilor dumneavoastră. Există multe, multe opțiuni, cele mai cunoscute fiind furnizate de Macromedia Flash și Shockwave, Java și JavaScript, Visual Basic și VBScript, Real Audio și Real Video și o gamă în continuă creștere de alte opțiuni interactive.

OBSERVAȚIE

Informații suplimentare privind produsele Macromedia pot fi găsite la `http://www.macromedia.com/`. Pentru Java și JavaScript, vizitați situl de bază al Sun Microsystem, la `http://www.sun.com/`. Ca dezvoltator al limbajului Java, Sun oferă actualizări, legături și alte resurse ajutătoare celor interesați în folosirea lor. Microsoft este dezvoltatorul pentru Visual Basic și VBScript; mai multe informații pot fi găsite la `http://www.microsoft.com/`. La `http://www.realaudio.com/` pot fi explorate opțiunile de mediu ale firmei Progressive Networks (Real Audio și Real Video).

Navigarea

Ați aruncat deja o privire asupra tipurilor de navigare prin forum. Această secțiune recapitulează mediile folosite și vă oferă o imagine a codului și a programării actuale în HTML, necesare pentru a realiza proiectul.

Mediile de navigare

În forumul DisAbilities sunt două tipuri de medii de navigare:

- **Bazate pe text:** Cele mai simple – dar adesea cele mai efective – opțiuni pentru siturile Web, opțiunile de navigare la nivel înalt, sunt codate după cum urmează:

```
<a href="contents.htm">Contents</a>¦
<a href="info.htm">Information</a>¦
<a href="whatsnew.htm">What's New</a>¦
<a href="chatmain.htm">Chats</a>¦
<a href="bbsmain.htm">BBSs</a>¦
<a href="libmain.htm">Files of Interests</a>¦
<a href="linksmain.htm">Related Sites</a>
```

SFAT

Observați că am folosit simbolul ¦ ca separator între selecții. Alte separatoare obișnuite pentru text sunt linia de despărțire (-) sau parantezele pătrate ([ ]). Oricare metodă folosiți, introducerea unei separări vizuale între opțiuni vă este de ajutor.

- **Macromedia Flash:** În cazul barei de navigare pe marginea din stânga, Microsoft Network oferă proiectanților posibilitatea ca navigarea să fie folosită la nivelul superior al tuturor siturilor interne.

Bara de navigare Flash este apoi codată folosind eticheta `<SCRIPT>`, care manipulează limbajul VBScript, care controlează la rândul său navigarea și permite ca aceasta să fie personalizată de la un forum la altul, și eticheta `<OBJECT>`, care gestionează fișierul Flash. (eticheta `<OBJECT>` este prezentată în Capitolul 11, „Adăugarea imaginilor pe pagina dumneavoastră Web" și 16, „Obiecte înglobate în HTML"; modul cum este folosită în această aplicație eticheta `<SCRIPT>` este descris în Capitolul 24, „Folosirea limbajului VBScript"). Codul este următorul (observați modul de înglobare a adreselor URL în cod):

```
<script language="vbscript">

sub navbar_fscommand(byval command, byval args)
select case command
case "show_all" parent.location.href=http://forums.msn.com"
```

```
case "home" parent.forummain.location.href="welcome.htm"
case "contents" parent.forummain.location.href="contents.htm"
case "info" parent.forummain.location.href="info.htm"
case "new" parent.forummain.location.href="whatsnew.htm"
case "chat" parent.forummain.location.href="chatmain.htm"
case "bbs" parent.forummain.location.href="bbsmain.htm"
case "file" parent.forummain.location.href="libmain.htm"
case "guestbook" parent.forummain.location.href
➥="news>//msnnews.msn.com/msn.forums.disabilities.guestbook"
case "suggestions" parent.forummain.location.href="news://msnnews.msn.com/
➥msn.forums.disabilities.sugestbox"
case "sites" parent.forummain.location.href="linksmain.htm"
end select
end sub

<script>

<object
    id="navbar"
    classid="clsid:d27cdb6e-ae6d-11cf-96b8-444553540000"
    width="100%" height="100%">
    <param name="movie" value="images/navbar.spl">
    <param name="quality" value="high">
    <param name="loop" value="false">
    <param name="play" value="false">
    <param name="scale" value="showall">
    <param name="devicefont" value="true">
    <param name="salign" value="tl">
    <param name="menu" value="false">
    </object>
```

Rezultatul obținut este un sistem de navigație atractiv și static, care nu numai că își îndeplinește menirea principală de a purta vizitatorii de la o pagină la alta, dar ajută și la realizarea scopului interfeței, proiectarea consecventă.

Stilurile de navigare

De obicei se folosesc marginile din dreapta, stânga, de sus sau de jos ale unei anumite paginii pentru navigare. Nivelul cel mai înalt al forumului DisAbilities are avantajul opțiunilor stânga și jos folosind bara de navigare Flash (stânga) și navigarea prin text (partea de jos).

Revista Rolling Reporter folosește și partea din stânga pentru navigare, dar aceasta mai mult sub formă de text decât grafică. Și în sfârșit festivalul Creative Arts folosește peste tot un sistem de navigare foarte simplu, în partea de jos a ecranului (vezi Figura 35.13).

În acest sit sunt folosite preponderent partea din stânga și de jos pentru navigare, dar în mod diferit. În cazul nivelului cel mai înalt, sunt utilizate cadrele fără chenar, pentru a realiza secțiunea statică din stânga, unde se află bara de navigare Flash. Pe de altă parte, Rolling Reporter folosește un format bazat pe tabele pe marginea din stânga.

Metodele de navigare

Vă voi prezenta pe scurt codul pentru diferite stiluri de navigare.

■ **Navigarea folosind cadrele**: Navigarea folosind cadrele la nivelul cel mai înalt folosește trei fișiere pentru a crea interfața proprie:

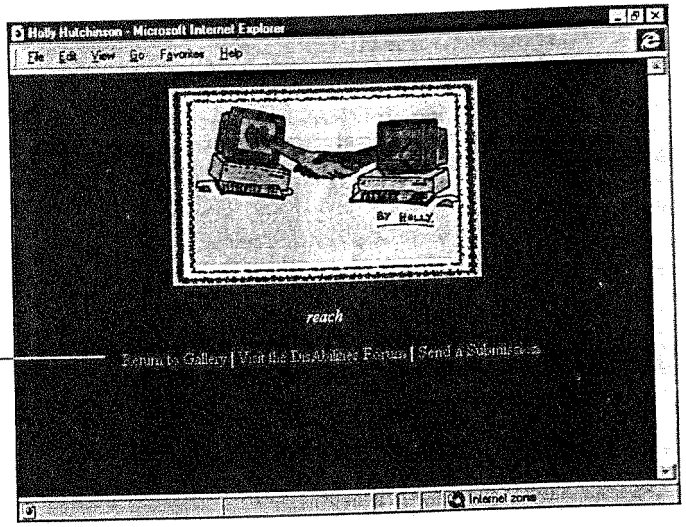


FIGURA 35.13. *Folosirea părții de jos a ecranului pentru o variantă de navigare simplă în cazul expoziției festivalului Creative Arts.*

Folosirea părții de jos a ecranului pentru navigare

Setul de cadre Acesta este fișierul „de control" al oricărui proiect care folosește cadrele; codul acestuia e prezentat aici:

```
<html>
<head>
<title>DisAbilities Forum</TITLE>
</head>

<frameset cols="95,20%" framespacing="0" frameborder="0">
  <frame src="sidebar.htm" name="forum_sidebar" scrolling="no">
  <frame src="welcome.htm" name="forummain" noresize>
</frameset>

</html>
```

Codul HTML precizează formarea a două coloane, cea din stânga lată de 95 de pixeli. Cadrele nu au chenar, iar defilarea e dezactivată. Acest set simplu de cadre configurează machetarea pentru paginile care se vor încărca pe coloanele din stânga și dreapta.

Primul cadru (cadrul din stânga) Acesta este cadrul care conține bara de navigare, cu scriptul prezentat mai devreme în acest capitol. Iată întreaga pagină:

```
<html>

<head>

<title>the disabilities forum </title>

<script language = "vbscript">

sub navbar_fscommand(byval command, byval args)
select case command
case "show_all" parent.location.href=http://forums.msn.com"
case "home" parent.forummain.location.href="welcome.htm"
case "contents" parent.forummain.location.href="contents.htm"
case "info" parent.forummain.location.href="info.htm"
case "new" parent.forummain.location.href="whatsnew.htm"
case "chat" parent.forummain.location.href="chatmain.htm"
```

```
case "bbs" parent.forummain.location.href="bbsmain.htm"
case "file" parent.forummain.location.href="libmain.htm"
case "guestbook" parent.forummain.location.href =
➥"news>//msnnews.msn.com/msn.forums.disabilities.guestbook"
case "suggestions" parent.forummain.location.href="news://
➥msnnews.msn.com/msn.forums.disabilities.sugestbox"
case "sites" parent.forummain.location.href="linksmain.htm"
end select
end sub

<script>

</head>

<body
<bgcolor="#FFFFFF" leftmargin="0" topmargin="0">

<object
  id="navbar"
  classid="clsid:d27cdb6e-ae6d-11cf-96b8-444553540000"
  width="100%" height="100%">
  <param name="movie" value="images/navbar.spl">
  <param name="quality" value="high">
  <param name="loop" value="false">
  <param name="play" value="false">
  <param name="scale" value="showall">
  <param name="devicefont" value="true">
  <param name="salign" value="tl">
  <param name="menu" value="false">
</object>

  </body>
  </html>
```

Acest cod HTML încarcă în cadrul din stânga navigarea Flash și apelurile pentru navigarea personalizată.

Al doilea cadru (cadrul din dreapta) Acesta reprezintă pagina de întâmpinare, pagina de început, principală a forumului, pe care ați văzut-o mai devreme în Figura 35.1. Veți vedea butoanele Flash, textul și alte elemente, inclusiv foile de stil, pe care le voi descrie în secțiunea „Spațiul, forma și fonturile", care urmează. Este interesant de observat macheta realizată cu ajutorul tabelului pentru controlul optim al paginii. Iată codul HTML pentru pagina de întâmpinare:

```
<html>
<head>

<title>

Welcome Page - AbleWorld: The DisAbilities Forum on MSN.

</title>

</head>

<style>

BODY {background: FFFFFF; color: 000000}
H3 {font: 14pt arial; color: 60099}
.1 {font: 11pt arial; color: 000000; text-align: right}
.2 {font: 12pt arial; color: 000000; text-align: left}
.3 {font: 10pt arial; color: 000000; text-align: left}
.4 {font: 11pt arial; color: 9966FF; text-align: right}
```



```
.5 {font: 8pt arial; color: FF9933; text-align: center}
A.p {color: FF9933; text-decoration: none}

</style>

<body bgcolor="#FFFFFF" text="#FFFFCC" link="#FFCC99"
vlink="#9999CC" alink="#FFFFCC" background="images/bak2.jpg">

<table border=0 width=510 cellpadding=5 cellspacing=5>
<tr>

<td class=3 valign=top width=200>
<OBJECT
classid="clsid:D27CDB6E-AE6D-11cf-96B8-444553540000"
codebase="http://active.macromedia.com/flash2/cabs/SWFLASH.CAB"
width="104" height="70">
<param NAME="Movie" VALUE="images/icon-wht.swf">
<param name="Loop" value="False">
<param NAME="Quality" VALUE="best">
<param name="Scale" value="showall">
</OBJECT>
<p>

<a class=p href="arts/default.htm" target=_top>Creative Arts Festival:</a>
<br>
<i>new exhibits weekly!</i>
</td>

<td class=1 valign=top width=300>
<img src="images/dis-ani1.gif" width=300 height=100 border=0 alt=" ">

</td>
</tr>

<tr>
<td class=3 width=500 colspan=2>

<object classid="clsid:D27CDB6E-AE6D-11CF-96B8-444553540000"
➥codebase="fsplash.cab" align="baseline" border="0" width="175"
➥height="35">
<param name="Movie" value="rr-but.swf">
<param name="Loop" value="False">
<param name="Play" value="True">
<param name="BGColor" value="ffffff">
<param name="Quality" value="high">
<param name="Scale" value="Showall">
</object>
<br>

<b>NO FEAR.</b> Rolling reporter Kelly Jones doesn't know the word
➥limitation. Clic the header, <b>read</b> this 'zine and neither will you.

</td>
</tr>

<tr>
<td class=3 width=100>

<object classid="clsid:D27CDB6E-AE6D-11CF-96B8-444553540000"
➥codebase="fsplash.cab" align="baseline" border="0" width="83" height="35">
<param name="Movie" value="rr-but.swf">
```

```
<param name="Loop" value="False">
<param name="Play" value="True">
<param name="BGColor" value="ffffff">
<param name="Quality" value="high">
<param name="Scale" value="Showall">
</object>
<br>
CWBailey:
<br>
Inner Vision
<br>
<a class=p href="http://forums.msn.com/disabilities/arts/bails-1.htm"
➥target="_top">Creative Arts on MSN!</a>

</td>

<td class=3 width=400>

<object classid="clsid:D27CDB6E-AE6D-11CF-96B8-444553540000"
➥codebase="fsplash.cab" align="baseline" border="0" width="70" height="35">
<param name="Movie" value="evt-but.swf">
<param name="Loop" value="False">
<param name="Play" value="True">
<param name="BGColor" value="ffffff">
<param name="Quality" value="high">
<param name="Scale" value="Showall">
</object>
<br clear="right">
<b>New! </b> Our <a class= p href="tip/daymain.htm" target=_top>Tip o' the
➥Day </a> helps inform you about a variety of DisAbilities services and
➥information. New tip every day!
<p>
</td>
</tr>

<tr>
<td valign=top width=500 colspan=2>

<div class=5>

<a href="contents.htm">Contents</a> |
<a href="info.htm">Information</a> |
<a href="whatsnew.htm">WHAT's New</a> |
<a href="chatmain.htm">Chats</a> |
<a href="bbsmain.htm">BBSs</a> |
<a href="libmain.htm">Files of Interests</a> |
<a href="linksmain.htm">Related sites</a>

</div>

</tr>
</table>

</body>
</html>
```

Cele trei fișiere combinate realizează interfața de navigare din stânga care domină modelul pentru nivelul superior al forumului DisAbilities.

■ **Navigarea pe baza tabelelor:** În situl revistei Rolling Reporter, navigarea se realizează în partea stângă, cu ajutorul unui tabel. Iată codul din pagina de contact a sitului revistei Rolling Reporter:

```
<html>
<head>
<title> Rolling Reporter: Contact</title>
</head>

<body bgcolor="#000000" text="#000000" link="#FF9933" vlink="#9966FF"
➥alink="9966CC" background="images/rr-bak.gif">

<table width=600 cellpadding=0 cellspacing=20 border=0>
<tr>
```

Această celulă a tabelului creează navigarea care folosește marginea din stânga:

```
<td valign=top align=right bgcolor="#000000" width="110>

<div align=center>
<img src="images/globe.gif" alt="globe graphic" width=100 height=109>
</div>
<p>

<font face=arial,helvetica size=3>

<a href="about.htm">about the rolling reporter</a>

<p>

<a href="articles.htm">travel articles</a>

<p>

<a href="sports.htm">sport and fitness</a>

<p>

<a href="access.htm">accesibility reports</a>

<p>

<a href="contact.htm">contact</a>

<p>

<a href="default.htm">home</a>

</font>

</td>
```

Această celulă a tabelului este celula principală de conținut:

```
<td valign=top width=450>

<div align=right>
<img src="images/ct-hed.jpg" width=300 height=50 border=0 alt="contact
➥header">
</div>
<p>

<font face=arial,helvetica>

Want to get in touch? Have an article, link, or other item of interest for
The Rolling Reporter Web site?
Send email to <a href="mailto:kelly_df@MSN.com">the rolling reporter</a>.

<p>

The Rolling Reporter holds live events and discussions on the Microsoft
Network's DisAbilities Forum.
```

```
You can follow <a href="http://forums.MSN.com/disabilities/">this link</a> to
access the Forum and learn about all of our activities!
<p>
The Rolling Reporter is the concept of Kelly Johns.
<p>
All articles and photos are &copy; 1997, Kelly Johns unless otherwise noted.
Reprints are restricted, please write for permission for article reprints.
Design and graphics are &copy; 1997 The DisAbilities Forum.
<p>
</font>
</td>
</tr>
</table>
<p>
<div align=left>
<font size=-1>&copy; 1997<br> the rolling reporter</a>
</div>
</body>
</html>
```

În acest moment, v-ați putea întreba care metodă este mai bună – cea cu tabele sau cea cu cadre. Cred că totul se rezumă la ce doriți să realizați. Tabelele se încarcă mai repede și ocupă mai puțin spațiu, deoarece există numai un fișier pe o pagină, iar cadrele necesită cel puțin trei fișiere pentru a se realiza o pagină. Totuși, cadrele pot aduce secțiunile statice în situl dumneavoastră, o metodă puternică și stabilă de proiectare a interfeței. Alegerea depinde de preferințele personale și de aplicație.

Lucrul cu culorile și grafica

Culoarea reprezintă o parte importantă a modului în care secțiunile forumului DisAbilities păstrează aspectul integrat.

Paleta personală

Întregul forum este dominat de o paletă de culori, formată din alb și negru cu o gamă de purpuriu, auriu și roșu. *Utilizarea* culorii diferă în mod substanțial, dar aceste culori sunt păstrate în mod consecvent în întreg situl.

Când se proiectează situri, este bine să folosim o paletă principală care să domine întregul sit, chiar dacă dumneavoastră folosiți culorile în mod diferit. De exemplu, dacă aveți la dispoziție alb, negru, roz, purpuriu, galben și verde în paleta dumneavoastră, puteți crea pe marginea din stânga un sistem de navigare care folosește diferite culori în zona de navigare, în funcție de locul din sit unde sunteți. Aspectul va fi diferit, dar paleta similară va contribui la consecvență. Este o cale de realizare a unei structuri orizontale interesante, fără a periclita consecvența și orientarea.

Culorile protejate

Întotdeauna recomand folosirea culorilor protejate ale browserului, subiect discutat în Capitolul 33, „Folosirea culorilor și a graficii în Web". Forumul DisAbilities nu este o excepție. Toate culorile folosite pentru fundaluri, text și grafică au fost preluate din culorile protejate ale browserului.

Iată eticheta <BODY> pentru nivelul cel mai înalt al forumului:

```
<body bgcolor="#FFFFFF" text="#FFFFCC" link="#FFCC99"
vlink="#9999CC" alink="FFFFCC" background="images/bak2.jpg">
```

Iată codul pentru Rolling Reporter:

```
<body bgcolor="#000000" text="#000000" link="#FF9933" vlink="#9966FF"
➥alink="#9966CC" background="images/rr-bak.gif">
```

și iată codul pentru una din paginile festivalului Creative Arts:

```
<body bgcolor="#FF9900" text="#000000" link="#CC3300" vlink="#663333"
➥alink="#FFFFCC" background="images/af-p1bak.gif">
```

Toate aceste culori sunt culori de browser protejate. Ca o mică experiență, puteți găsi aceste culori și să alcătuiți paleta forumului dumneavoastră singur. În Capitolul 33 găsiți informații privitoare la locul unde puteți găsi resursele pentru a face acest lucru.

Grafica

Toată grafica din nivelul cel mai înalt al sitului este constituită din fișiere GIF. Există un motiv întemeiat pentru aceasta: pentru a combina fundalul cu grafica, a fost important să fie folosită transparența. După cum se vede în figura 35.14, chiar și grafica animată de pe pagina de întâmpinare este transparentă.

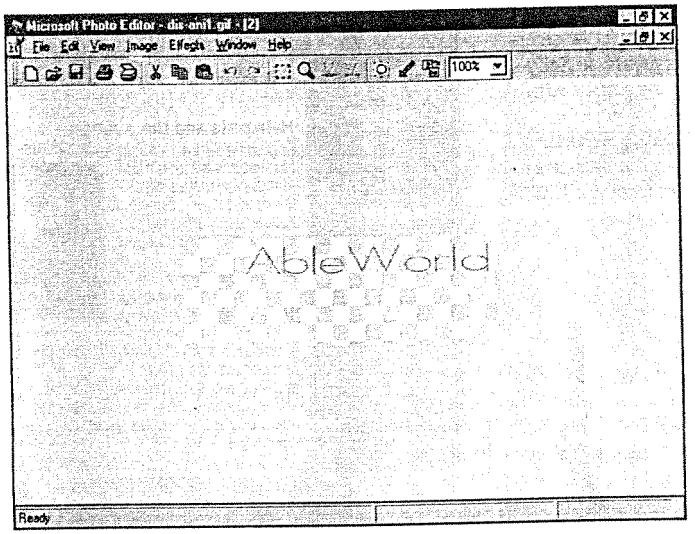


FIGURA 35.14. *Acest fișier GIF animat este și el transparent.*

Cu excepția fotografiilor fețelor din zona destinată personalului, nu s-au folosit fotografii cu lumină variabilă în secțiunea principală a forumului. Totuși, festivalul Creative Arts are mai multe expoziții de fotografii (vezi Figura 35.15), ca și revista Rolling Reporter, după cum se vede în Figura 35.16. În aceste cazuri, calitatea fotografiilor cere într-adevăr utilizarea formatului JPG.

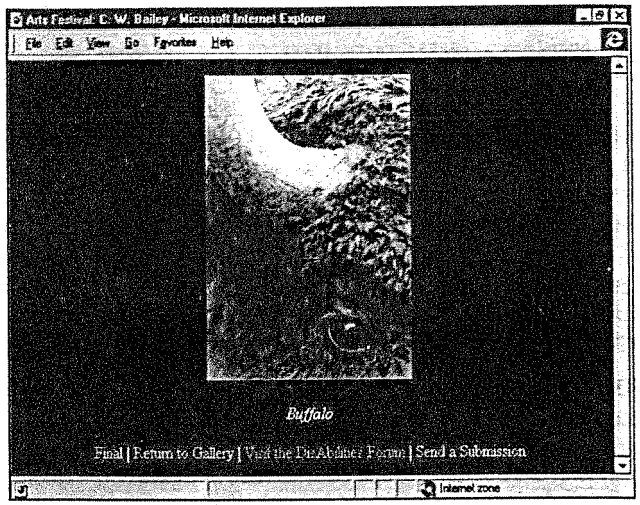


FIGURA 35.15. *Această fotografie făcută de C.W. Bailey a avut nevoie de calitatea oferită de formatul JPG.*

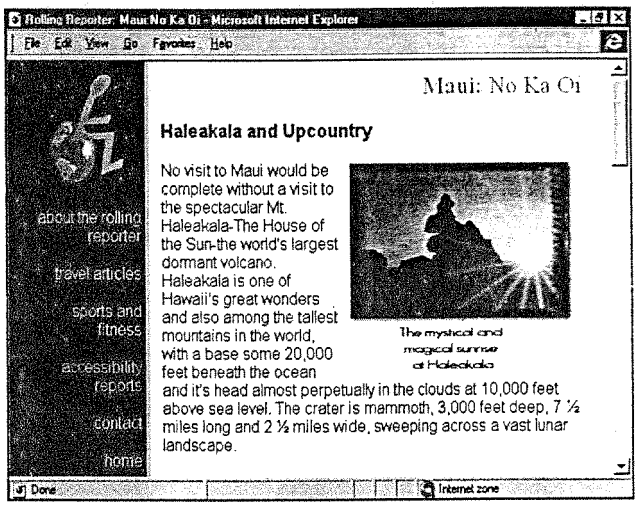


FIGURA 35.16. *Fotografia de excepție făcută de Kelly Johns pentru situl Web al revistei Rolling Reporter a solicitat de asemenea formatul JPG.*

Spațiul, forma și fonturile

Aspectele importante ale proiectării, referitoare la spațiu, formă și fonturi sunt tratate în forumul DisAbilities în diferite feluri.

În primul rând, spațiul este abordat prin folosirea tabelelor în toate paginile acestui sit. Chiar și într-o pagină simplă, cum este indexul festivalului Creative Media, se folosesc tabelele pentru organizarea pe coloane, dar și pentru spațiu:

```
<html>
<head>
<title>Arts Festival: The Artists</title>
```



```
</head>
<body bgcolor="#000000" text="#FFFFCC" link="#FF9900" vlink="CC3300"
➥ alink="#FFFFCC"

<table border=0 cellpadding=10 cellspacing=10 width=585>
<tr>
<td valign=top width=160>
<font face=arial color="#CC3300">v o i c e s</font>
<p>

<font face=arial size=2>

<b>Scott Reynolds: <a href="sr-1.htm"><br></b>A Tale of Madness (poem)</a>
<p>

<b>Scott Reynolds: <a href="schiz1.htm"><br></b>Schizophrenia (poem)</a>
<p>

<b>Barry Michaels: <a href="bm.htm"><br></b>Poems</a>
<p>

<b>April: </b> <br><a href="april.htm"> Happy Birthday (song)</a>
<p>

<b>Bill Schmidter: <a href="bill1.htm"><br></b>The rose (poem)</a>
<p>

<b>David Elmirage: <a href="david1.htm"><br></b>Friends (poem)</a>
<p>

<b>Rhea Adams: <a href="ra-1.htm"><br></b>Storm (poem)</a>
<p>

<b>Justin Roll> </b><br><a href="jr.htm"> Poems</a>
<p>

</font>
</td>

<td width=2 valign=top>
<img src="images/org-bar.gif" alt="orange bar" width=2 height=300>
</td>

<td valign=top width=160>
<font face=arial color="#CC3300">v i s i o n</font>
<p>

<font face=arial size=2>
<b>CW Bailey:</b>
<br><a href="bails-1.htm">Inner Vision (photos)</a>
<p>

<b>Kelly Johns:</b> <br><a href="kelly.htm">Words and Images
(photos and words)</a>
<p>

<b>Holly Hutchinson:</b>
<br><a href="hol1.htm">Then, Now, Someday (photos and words)</a>
<p>

<b>Holly Hutchinson:</b>
<br><a href="hol2.htm">Reach (drawn art)</a>
<p>

<b>Jerry Lemmon:</b>
```

```
<br><a href="jer-1.htm">Alaskan Flowers (photos)</a>
<p>
</font>
</td>
<td width=2 valign=top>
<img src="images/org-bar.gif" alt="orange bar" width=2 height=300>
</td>
<td valign=top width=160>
<font face=arial color="#CC3300">v i s t a s</font>
<p>
<font face=arial size=2>
<b>Nancy Lemmon:</b>
<br><a href="nancy.htm">Dolls (craft work)</a>
<p>
<b>Jeremy and Joshua Roberts:</b>
<br><a href="jj.htm">DisAbilities Forum Tribute (animation)</a>
</font>
<p>
</td>
</tr>
</table>
<div align=center>
<a href="http://forums.MSN.com/disabilities/">Visit the DisAbilities Forum</a>
➥| <a href="maito:kelly_df@MSN.com"">Send a Submission </a>
</div>
</body>
</html>
```

Observați dungile portocalii, create cu ajutorul fișierelor GIF și întinse pe verticală. Această metodă este convenabilă pentru crearea spațiului, atât pe orizontală, cât și pe verticală. Puteți face acest lucru folosind culoarea sau grafica transparentă, pentru a crea o distanțare insesizabilă.

O altă cale de a câștiga spațiu este alinierea imaginilor. Iată un exemplu HTML din pagina Alaskan Flowers (Flori din Alaska), a lui Jerry Lemmon:

```
<img src="images/jer-f12.jpg" alt="another flower" width=125 height=151
➥hspace=5 vspace=10 align=right>
```

În acest caz, puteți vedea modul cum se folosește alinierea, spațiul pe orizontală (hspace) și cel pe verticală (vspace) pentru a furniza mai mult spațiu în machetă.

Forma este întrebuințată în multe locuri. Ați observat probabil tema arcului, care parcurge paginile de la cel mai înalt nivel din forum. O altă utilizare a formei poate fi găsită la intrarea în situl festivalului Creative Arts (vezi Figura 35.17) și pe tot parcursul expoziției festivalului, după cum se vede în Figura 35.18. Forma stimulează mult interesul pentru forum, și la fel o va face și pentru proiectul dumneavoastră!

În final, tipul caracterelor reprezintă o parte integrantă a consecvenței sitului. Cu excepția festivalului Creative Arts, unde s-au folosit tipuri mai puțin convenționale de caractere, tipurile, culorile și mărimile fonturilor sunt specifice. Se folosește același tip de font, denumit Espresso, pentru anteturile paginilor de pe nivelul cel mai înalt și pentru revista Rolling Reporter, după cum puteți vedea din comparația celor două anteturi, în Figura 35.19. Corpul textului folosește un tip fără serife, denumit Arial.

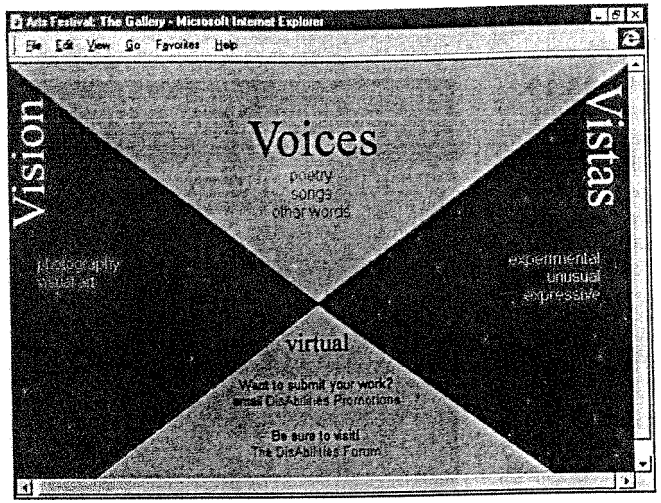


FIGURA 35.17. *Pentru intrarea în situl festivalului Creative Arts se folosește un stil neobișnuit, cu triunghiuri.*

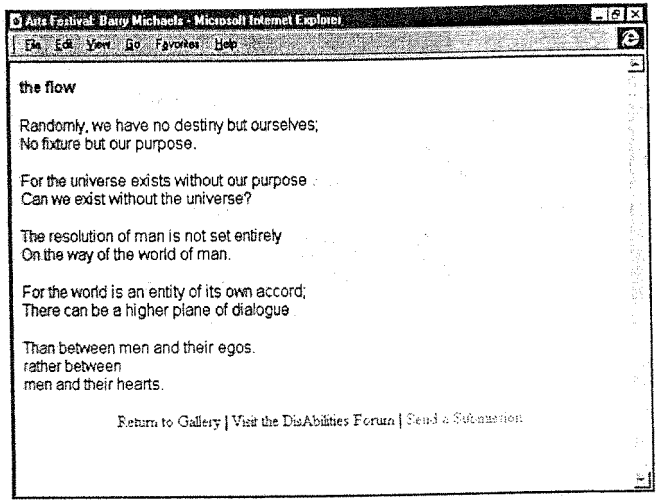


FIGURA 35.18. *Spirala accentuează aspectul emoțional al conținutului paginii.*

Veți observa că fonturile sunt controlate de foi de stil înglobate în paginile specifice ale forumului.

Iată un exemplu:

```
< style >
BODY {background: FFFFFF; color: 000000}
H3 {font: 14pt arial; color: 60099}
.1 {font: 11pt arial; color: 000000; text-align: right}
.2 {font: 12pt arial; color: 000000; text-align: left}
.3 {font: 10pt arial; color: 000000; text-align: left}
.4 {font: 11pt arial; color: 9966FF; text-align: right}
.5 {font: 8pt arial; color: FF9933; text-align: center}
```

```
A.p {color: FF9933; text-decoration: none}
</style>
```

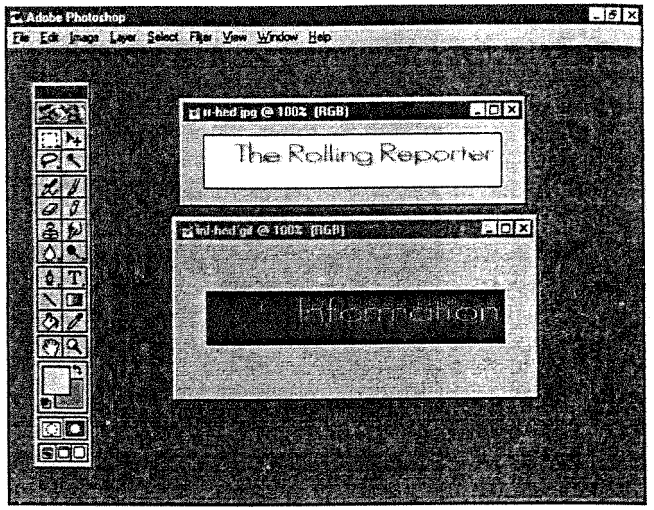


FIGURA 35.19. *Anteturile grafice pentru forum și revista Rolling Reporter folosesc același tip de font, numit Espresso.*

Pentru mai multe informații privitoare la foile de stil, consultați capitolul 18, „Utilizarea foilor de stil în cascadă"

NOTĂ

Puteți obține aceleași rezultate pentru fonturile din paginile dumneavoastră, folosind eticheta <FONT> și atributele corespunzătoare. Deoarece toată lumea folosește Internet Explorer în Microsoft Network, folosirea foilor de stil în proiectare nu pune probleme potențiale de vizualizare pentru această concepție a sitului. Totuși, trebuie să cântăriți argumentele pro și contra utilizării foilor de stil, a etichetei <FONT> și a atributelor pentru fiecare proiect în parte.

Rezumat

Această examinare detaliată a unui sit proiectat și implementat în mod profesional vă oferă un exemplu de lucru cu procedeele pe care le-ați învățat în ultimele patru capitole. E important să nu uitați că proiectarea interfeței, realizarea graficii și lucrul cu culoarea, spațiul și fonturile trebuie să fie toate precedate de o examinare preliminară detaliată a audienței și scopului dumneavoastră. Fără acestea, calea pe care mergeți va fi marcată de hazard, în cel mai bun caz. Totuși, dacă ați făcut lucrurile de bază, ați studiat atent procedeele din aceste capitole și ați urmărit modul în care aceste procedee s-au aplicat forumului DisAbilities din Microsoft Network, sunteți capabil să începeți să folosiți aceste procedee. Rezultatele vor fi sigur situri Web mai bine gândite, mai bine concepute și – în final –încununate de un succes mai mare.

PARTEA a VIII-a

XML

Capitolul 36
Ce este limbajul XML?

de Simon North

În acest capitol:

- **Ce este limbajul XML? 564**
- **Documente XML 564**
- **Regulile limbajului XML 566**
- **Declararea documentelor XML 566**
- **Declararea tipului documentului XML 567**
- **Legăturile XML 572**
- **Stilul XML 578**
- **Viitorul limbajului XML 580**

Limbajul XML, ale cărui inițiale provin de la *Extensible Markup Language* (Limbaj extins de marcare), reprezintă un dialect foarte simplu al limbajului SGML (*Standard Generalized Markup Language* – Limbaj standard de marcare generalizat). Limbajul XML a fost dezvoltat pentru a umple golul dintre puterea și complexitatea limbajului SGML și deficiențele și simplitatea limbajului HTML

Acest capitol cuprinde următoarele aspecte:

- Scopul și natura limbajului XML
- Regulile de sintaxă și structură ale limbajului XML
- Declarații ale tipului documentelor XML (DTD)
- Mecanismele pentru legături ale limbajului XML (XLL)
- Limbajul de stil al XML (XSL)

După ce citiți acest capitol, ar trebui să cunoașteți conceptele de bază ale limbajului XML și să înțelegeți bazele limbajului de stil asociat XML, XSL.

NOTĂ

Aveți grijă că limbajul XML nu este definitivat încă. Specificațiile conțin trei părți, fiecare parte fiind într-un stadiu diferit. Partea I, care acoperă limbajul de bază, se află acum la a treia revizuire, dar este încă în faza de proiect. Partea a-II-a, care tratează legăturile, se află la a doua versiune a proiectului. Partea a-III-a, care acoperă limbajul de stil, este un proiect necorectat care practic nu are o formă definită. Deși specificația e relativ stabilă, pot apărea modificări ale oricărei părți a limbajului XML.

Ce este limbajul XML?

După cum am spus, XML este un dialect foarte simplu al limbajului SGML, așa că, pentru a-l înțelege corespunzător, trebuie să cunoașteți mai întâi câteva lucruri despre SGML.

În principiu, SGML este un limbaj care furnizează instrumentele pentru definirea limbajelor de marcare. HTML este unul dintre limbajele de marcare, denumit *aplicație* a limbajului SGML, care constă dintr-un set de elemente bine definit (conținute în *etichete*), atribute (de exemplu `color`) și entități (de exemplu `©`).

De la formularea ca standard ISO, în anul 1986, limbajul SGML a fost confruntat permanent cu problema propriei creații. Încercând să fie un limbaj deplin generalizat, trebuia să fie puternic. Ca să fie puternic, a devenit complex și, devenind complex, a devenit greu de învățat, consumator de timp la implementare și scump.

Internetul (sau mai precis World Wide Web) a avut nevoie de ceva mai simplu, mai puțin solicitant pentru resursele umane și ale calculatorului – și așa a apărut limbajul HTML.

HTML a fost un limbaj de marcare excelent pentru scopul în care a fost creat, dar în timpul scurs de la realizarea lui, acest scop a evoluat într-o manieră nebănuită de nimeni. HTML este acum pus la lucru într-o manieră care a ajuns la limitele posibilului și, în pofida încercărilor de a-l îmbunătăți cu extensii particulare, se pare că și-a epuizat resursele. Aplicațiile Web moderne necesită mai mult decât ceea ce poate oferi simpla adăugare a câtorva etichete.

Limbajul XML este o încercare de a găsi un domeniu comun între SGML și HTML. În loc să fie o altă aplicație a limbajului SGML, adică un limbaj asemănător cu HTML, XML este – precum SGML – un „metalimbaj" pentru definirea limbajelor de marcare. Aplicațiile XML reprezintă o trusă de instrumente care vă ajută să creați propriul dumneavoastră limbaj HTML. Puteți folosi XML pentru a defini propriile dumneavoastră aplicații sau propriul dumneavoastră limbaj de marcare, astfel încât să puteți coda informațiile în documentele Web mult mai precis decât o puteți face în limbajul HTML. Combinați limbajul XML cu puterea de procesare pe care o oferă limbaje precum Java și ActiveX și ați găsit calea de a realiza documente „inteligente", care pot fi înțelese de aplicațiile dumneavoastră în așa fel încât informațiile pot fi procesate folosind metode total neaccesibile limbajului HTML.

XML nu are ca scop înlocuirea HTML; el este conceput ca un instrument adițional și o soluție alternativă la HTML, ori de câte ori este nevoie. XML și HTML pot coexista în pace, fiecare fiind folosit pentru ceea ce este mai potrivit: HTML pentru aplicațiile mai puțin pretențioase, cum sunt paginile Web, iar XML pentru aplicațiile care necesită documente mai inteligente și posibilități mai mari de procesare.

Documente XML

Un document XML nu trebuie să arate foarte diferit față de un document HTML (și, după cum veți vedea mai târziu în acest capitol, nu e nevoie de un efort prea mare pentru a converti un document HTML într-unul XML). Ca și documentele HTML, un document XML reprezintă un amestec de text și marcaje (etichete). Documentele XML diferă de cele HTML la nivelul conceptual – în special în ceea ce privește validitatea și formatul corespunzător.

Validitatea

Ca o reflectare a relației mai strânse cu SGML decât cu HTML, documentele XML pot fi validate pentru a avea siguranța că structura și conținutul respectă regulile definite pentru acel tip de document. Ca și în cazul limbajului SGML, aceste reguli sunt precizate într-un document numit *declarație a tipului documentului* (document type declaration – DTD).

Spre deosebire de alte sisteme de marcare (cum sunt HTML și TeX), folosirea declarației DTD în SGML sau XML, semnifică de fapt verificarea sau, mai bine zis, validarea documentelor. *Validarea* reprezintă faptul că puteți verifica dacă toate elementele pe care ar trebui să le conțină un document sunt de fapt existente, în ordinea firească și, până la o anumită limită, dacă conțin datele pe care ar trebui să le conțină.

Voi reveni mai târziu la declarația DTD în acest capitol, la subcapitolul „Declararea tipului documentului XML", dar pentru moment este suficient să înțelegem că un document XML este „valid" dacă are o declarație DTD, iar conținutul documentului respectă regulile precizate în acea declarație DTD. Regulile specifice pentru validitate sunt explicate în subcapitolul următor, pe măsură ce sunt descrise obiectele cărora le sunt aplicate.

Formatul corespunzător

Pentru a salva capacitatea de transmitere a datelor pe Internet, pentru a reduce necesarul de procesare (și implicit viteza de interpretare) și pentru a ține cont de condițiile de transport necunoscute și imprevizibile, nu este rezonabil – și nici posibil chiar – să fie întotdeauna disponibilă o declarație DTD. S-ar putea să nici nu doriți tipul controlului pe care l-ar impune validarea (în definitiv, browserele Web nu verifică valabilitatea documentelor HTML). Pentru a lua în considerare transmiterea „fără restricții" a documentelor XML, s-a renunțat la o parte importantă formalitatea limbajului SGML și XML a introdus noul concept de *format corespunzător*.

Documentele XML cu format corespunzător nu necesită o declarație DTD (de aceea reputația XML de a fi „un SGML fără DTD" nu e de fapt corectă). Chiar dacă declarația DTD există, documentele XML nu au nevoie ca referința la declarația DTD, să fie procesată corect (mediul XML este conceput ca un strat cu două etaje, având un modul procesor XML care citește documentele aplicației și dă acces la conținutul și structura acestora). Bineînțeles, pentru ca un document XML să fie procesat corect fără referința la informațiile privind structura sa, care ar fi existat în DTD, sunt necesare anumite concesii. Documentele XML cu format corespunzător trebuie să aibă următoarele caracteristici:

- Să înceapă cu o declarație de marcare necesară (amănunte mai târziu), care să precizeze că nu este necesară declarația DTD, cum ar fi:

```
<?XML VERSION="1.0" RMD="NONE" ?>
```

- Să conțină unul sau mai multe elemente
- Să conțină toate elementele imbricate unul în interiorul celuilalt și toate etichetele să fie echilibrate (pentru orice semn de început de etichetă trebuie să existe un semn de sfârșit de etichetă).
- Toate atributele să fie între ghilimele.
- Etichetele de început ale elementelor lipsite de conținut să fie închise cu șirul special `/>` sau dacă au conținut prin adăugarea unor etichete de închidere specifice (cum este `<HR></HR>`).
- Să nu conțină nici-un caracter de marcare printre caracterele care reprezintă datele (caracterele de marcare trebuie să fie incluse în secvența escape folosind entități definite anterior).
- Toate atributele elementelor sunt de tip `CDATA` (date caracter) deoarece nu există altă cale să le definim altfel fără declarația DTD.

Aceste restricții de formatare corespunzătoare vor avea sens în mai mare măsură în momentul în care veți afla despre obiectele cărora li se aplică. Totuși, înainte de a intra prea mult în detaliile privind validitatea și formatul corespunzător, să vedem câteva din regulile de bază care se aplică documentelor HTML.

Regulile limbajului XML

Un document XML conține o entitate denumită *entitatea documentului*, care reprezintă punctul de pornire al procesorului XML și poate conține întreg documentul, la fel cum elementul HTML conține un document HTML complet.

Entitățile trebuie să conțină fiecare un număr integral de elemente, comentarii, instrucțiuni de procesare și referințe, cu posibilitatea ca datele caracter să nu fie în nici un element al entității, altfel ele trebuind să conțină date nontext și nici un element.

Structurile logice și fizice (elementele și entitățile) dintr-un document XML trebuie să fie sincrone, ceea ce înseamnă că etichetele și elementele trebuie să înceapă și să se termine fiecare în aceeași entitate, deși pot face referiri din interior la alte entități.

Etichetele de început și etichetele de sfârșit

Eticheta de start marchează începutul oricărui element care are conținut. Eticheta de sfârșit, cu același nume ca al etichetei de început, marchează sfârșitul fiecărui element, de exemplu:

```
</TITLE>Simon's Home Page</TITLE>
```

Dacă un element nu are conținut, eticheta de start poate reprezenta întreg elementul, iar eticheta ia o formă specială: `<name/>`.

Comentariile

Comentariile folosesc aceeași sintaxă ca în limbajul HTML:

```
<!-- acesta este un comentariu -->
```

Totuși, comentariile trebuie scrise folosind *exact* sintaxa specificată. Comentariile trebuie să îndeplinească următoarele condiții:

- Este interzisă folosirea spațiului în șirurile de start (`<!--`) și de sfârșit (`-->`).
- Comentariile nu pot fi imbricate.
- Șirul -- (două linii de despărțire) nu trebuie să existe în comentariu.
- Comentariile pot apărea în orice loc, cu excepția secvențelor CDATA, a declarațiilor sau a etichetelor.

Declararea documentelor XML

Prima linie a unui document XML conține o declarație XML care specifică versiunea XML folosită:

```
<?XML VERSION="1.0" ENCODING="UTF-8" RMD="NONE" ?>
```

Observați folosirea semnului de întrebare; această instrucțiune este din punct de vedere tehnic o instrucțiune de procesare. Toate instrucțiunile de procesare ?XML sunt rezervate pentru utilizarea numai de către limbajul XML.

Declarația ENCODING identifică setul de caractere în care este codat documentul. Spre deosebire de HTML, care promovează setul ASCII, XML promovează Unicode, care permite caracterelor străine să fie folosite mult mai ușor decât în HTML. (Pentru mai multe informații asupra seturilor de caractere străine, vezi Capitolul 40, „Internaționalizarea limbajului HTML".)

Declarația de marcare cerută (required markup declaration – RMD) precizează dacă e necesară procesarea declarației DTD; prin aceasta puteți identifica dacă documentul dumneavoastră este (potențial) valabil sau „numai" formatat corespunzător. Valorile posibile ale declarației sunt următoarele:

- NONE: Procesorul XML poate analiza corect documentul conținut, fără să citească mai întâi nici o parte a declarației DTD.
- INTERNAL: Procesorul XML trebuie să citească și să proceseze subsetul intern al declarației DTD, dacă există.
- ALL (valoare predefinită): Procesorul XML trebuie să citească și să proceseze declarațiile din ambele subseturi ale declarației DTD, dacă acestea există.

Declararea tipului documentului XML

Declararea tipului documentului (DTD) XML conține declarații ale elementelor și atributelor care definesc structura elementelor permise ale unui document XML de un anumit tip.

Declarația DTD trebuie să fie așezată înaintea primei etichete de început din documentul respectiv, după cum urmează:

```
<!DOCTYPE letter SYSTEM "letter.dtd">
```

După cum arată această linie, declarația DTD se află în fișierul `letter.dtd`.

Partea declarației DTD care se află într-un fișier extern poartă numele de *subset extern* al DTD. Ca și în cazul foilor de stil CSS1, în care stilul poate fi definit extern sub forma de foi de stil, sau intern, în același document, o declarație DTD poate să includă în mod direct și un alt subset, intern:

```
<!DOCTYPE letter [
  <!ELEMENT address (#PCDATA)>

<!-- aceasta parte contine entitatea, elementul si atributul -->
<!-- declaratiilor, folosind aceeasi sintaxă ca o declaratie -->
<!-- externa a tipului documentului -->

]>
<letter>
<!-- urmeaza corpul documentului -->
</letter>
```

Cele două subseturi formează împreună declarația tipului documentului. Unul din subseturi poate să fie lipsit de conținut (sau pur și simplu absent), dar nu pot fi lipsite de conținut amândouă subseturile.

Dacă sunt utilizate ambele subseturi, cel intern este citit primul. Declarațiile referitoare la entitate și atribut din subsetul intern au prioritate față de cele din subsetul extern.

Numele din declarația DTD trebuie să coincidă cu numele elementului rădăcină din documentele XML de acel tip.

Următorul subcapitol examinează mai amănunțit conținutul declarației DTD pentru documentele XML.

Declararea elementelor

Fiecare tip de element care apare într-un document XML valid trebuie declarat o dată în declarația DTD. Declararea unui element restricționează tipul și conținutul elementului, după cum se vede în exemplul următor:

```
<!ELEMENT BOOK (FRONT,BODY, BACK)>
```

Prin declararea elementului se precizează că elementul BOOK conține un element FRONT urmat de BODY, care la rândul lui este urmat de elementul BACK. Toate trei elementele trebuie să fie prezente și să apară în această ordine.

Declarațiile privind elementele permise a fi folosite într-un alt element poartă numele de *modele de conținut*. Conținutul instanței unui element trebuie să coincidă cu modelul de conținut declarat pentru acel tip de element.

Limbajul folosit pentru specificarea modelelor de conținut al elementelor folosește următoarea sintaxă:

Tabelul 36.1. Sintaxa modelului de conținut.

Simbol	*Semnificație*
( )	Delimitează o secvență sau un set de variante.
,	Separă tipurile de elemente dintr-o secvență (se citește „urmat de"; `A,B` înseamnă „A urmat de B".
|	Separă tipurile de elemente alternative dintr-o listă de variante (`A | B` înseamnă „A sau B").
?	Urmează un element sau un grup de elemente și indică faptul că acesta apare o dată sau niciodată (`A,B?` înseamnă „A" sau „A,B,").
*	Urmează un element sau un grup de elemente și indică faptul că acesta apare de mai multe ori sau niciodată (`A,B*` înseamnă „A" sau „A,B,B...").
+	Urmează un element sau grup de elemente și indică faptul că acesta apare o dată sau de mai multe ori (`A,B+` înseamnă „A,B" sau „A,B,B,B...").

Conținutul unui element poate fi „conținut din elemente" (poate fi format numai din alte elemente) sau „conținut amestecat" (poate fi format din date caracter amestecate cu elemente).

Datele caracter sunt declarate ca fiind de tipul `#PCDATA` (date caracter analizate). Simbolul # (indicator de nume rezervat în SGML) face distincția între acest nume al tipului elementului și oricare alt tip al elementului numit PCDATA (care este un nume perfect legal și poate fi folosit oriunde). Datele caracter analizate (sau mai simplu date caracter, din punct de vedere al XML) trebuie să fie procesate pentru a se verifica dacă fiecare caracter este de marcare sau este un caracter simplu din conținutul datelor. Oricare caracter de marcare (etichete sau simboluri ale entităților) din datele caracter analizate va fi interpretat. Caracterele de marcare care nu sunt destinate interpretării trebuie *precedate de secvențe escape* (folosind referințele entităților) sau *ascunse* (folosind secțiunile `CDATA`).

Dacă un element are conținut amestecat, trebuie declarată mai întâi partea cu date caracter, iar modelul conținutului trebuie să folosească separatorul `OR`, astfel:

```
<!ELEMENT address (#PCDATA | street | number )>
```

Elementele care nu au conținut sunt declarate folosind cuvântul cheie `EMPTY`, dar verificați dacă într-adevăr ele sunt lipsite de conținut când apar în documentul XML.

Un element declarat prin folosirea cuvântului cheie `ANY` poate conține elemente copil de orice tip și număr, amestecate cu date caracter.

Atribute

Atributele asociază anumite proprietăți (mărime, culoare, aliniere ș.a.) unei clase de elemente. Deoarece adesea sunt declarate în același timp mai multe proprietăți, atributele sunt specificate în *declarații de listă de atribute*, care precizează numele, tipul datelor și valoarea prestabilită (dacă există) ale fiecărui atribut asociat cu un anumit tip de element, după cum se vede în exemplul din pagina următoare:


```
<!ELEMENT person (#PCDATA)>
<!ATTLIST person
  forename   CDATA      #REQUIRED
  surname    CDATA      #REQUIRED
  age        NUMBER     #IMPLIED>
```

Atributele propriu-zise pot apărea numai în etichete de deschidere, ca în exemplul următor:

```
<person forename="Fred" surname="FLINSTONE" age="5">
```

Când se declară mai multe liste de atribute pentru un anume tip de element, tot conținutul acestora e concatenat. De asemenea, când sunt date mai multe definiții pentru același atribut al unui anumit element, e luată în considerare numai prima declarație, iar următoarele sunt ignorate.

Remaparea atributelor

Pentru a evita conflictele dintre elementele a căror atribute au aceleași nume (ca de exemplu situația în care un subset intern DTD și unul extern declară aceleași atribute), atributele pot fi declarate ca fiind echivalente, folosind atributul `XML-ATTRIBUTES`.

Acest atribut trebuie să conțină perechi de nume. În fiecare pereche, primul nume trebuie să fie `ROLE`, `HREF`, `TITLE`, `SHOW`, `INLINE`, `CONTENT-ROLE`, `CONTENT-TITLE`, `ACTUATE`, `BEHAVIOR` sau `STEPS`. Cel de-al doilea nume este tratat ca și cum ar juca rolul repartizat primului.

Tipurile atributelor

Atributele XML sunt de trei feluri: tip șir, un set de tipuri token și tipuri enumerate.

Tipul șir poate lua ca valoare orice șir literal, dar tipul token are diferite restricții lexicale și semantice:

- Tipul `ID` trebuie să fie un simbol `NAME` valid. Numele nu trebuie să apară mai mult decât o dată într-un document XML ca `ID`.
- Valoarea `IDREF` trebuie să coincidă cu valoarea atributului `ID`, al unui element din documentul XML.
- Tipurile `ENTITY` și `ENTITIES` trebuie să coincidă cu numele unei entități binare generale externe, declarate în DTD.
- Tipurile `NMTOKEN` și `NMTOKENS` trebuie să fie un șir text conținând o literă sau o linie de subliniere (underscore), urmată de una sau mai multe litere sau cifre.
- Notațiile folosite pentru numele atributelor trebuie să coincidă cu una dintre notațiile pentru nume folosite în declarație.

Valorile tipurilor enumerate trebuie să coincidă cu una din valorile `NMTOKEN` ale tokenurilor din declarație. O valoare `NMTOKEN` nu trebuie să apară mai mult decât o dată în tipurile de atribute enumerate ale unui singur tip de element.

Valorile predefinite ale atributelor

Pentru a evita cazul în care atributele trebuie declarate în mod explicit pentru fiecare element, puteți declara valori prestabilite ale atributelor în DTD. Dacă este declarată o valoare prestabilită și atributul este omis, atributul este considerat ca și cum ar fi prezent, iar valoarea sa este cea prestabilită.

Declararea atributelor furnizează informații privitoare la necesitatea prezenței acestora, iar dacă prezența nu este necesară, modul cum trebuie să reacționeze procesorul XML dacă atributul declarat e absent în document:

- **`#FIXED` (fixat):** Documentul nu este valid dacă atributul are o valoare diferită de cea prestabilită.

- `#REQUIRED` (cerut): Documentul nu este valid dacă există o etichetă de început pentru tipul elementului care nu precizează o valoare pentru acest atribut.
- `#IMPLIED` (presupus): Dacă atributul este omis, procesorul XML trebuie să informeze pur și simplu aplicația că nu a fost precizată nici o valoare.

Dacă atributul nu este nici `#REQUIRED` (cerut), nici `#FIXED` (fixat), valoarea acestuia este cea declarată ca prestabilită.

Atributul pentru spațiu alb

În documentele HTML, spațiile libere multiple (numite *spații albe*) sunt reduse la un singur spațiu alb. Uneori, anumite spații albe au totuși o semnificație și dumneavoastră doriți să le păstrați. Atributul `XML-SPACE` poate fi folosit pentru identificarea elementelor care conțin spațiu alb cu semnificație pentru aplicații. Valoarea acestor atribute determină ce trebuie făcut cu spațiul alb din cadrul elementelor:

- `DEFAULT`: Semnifică faptul că e acceptată procesarea prestabilită a spațiului alb.
- `PRESERVE`: Semnifică faptul că tot spațiul alb trebuie păstrat.

Entități

Entitățile sunt similare tipurilor de macroinstrucțiuni pe care v-ați aștepta să le găsiți într-un pachet de programe pentru procesarea textului; ele sunt șiruri scurte de caractere care pot fi folosite ca prescurtări pentru pentru porțiuni de text (sau marcaje). Când sunt întâlnite de procesorul XML, aceste prescurtări sunt expandate. Entitățile pot fi binare sau text. O *entitate text* conține date sub formă de text, considerat ca fiind parte integrală a documentului. O *entitate binară* conține date sub formă binară cu o notație asociată (declarațiile privind notația vor fi explicate în subcapitolul „Entități externe").

Numai entitățile text pot fi referite cu referințe de entități și numai numele entităților binare pot fi date ca valori ale atributelor entităților.

Dacă se declară de mai multe ori aceeași entitate, se ia în considerare prima declarație întâlnită.

Într-un document XML cu format corespunzător, numele dat într-o referință a entității trebuie să coincidă cu numele dat în declarația entității. Referințele entității nu trebuie să conțină numele unei entități binare, iar entitățile binare pot fi apelate numai în cazul atributelor tip `ENTITY` sau `ENTITIES`.

Entitățile parametru

Referințele *entității parametru* pot fi utilizate într-un număr diferit de locuri din cadrul declarației DTD. Entitățile parametru vă oferă un mijloc eficient de refolosire a codului de marcare în mai multe locuri, astfel încât trebuie să tipăriți codul o singură dată. De exemplu, pentru a declara o entitate parametru, ar trebui să folosiți o declarație de felul următor:

```
<!ENTITY % font (em|tt|bold|strong)>
```

pe care o veți referi în felul următor:

```
<!ELEMENT body (para, %font;)*>
```

Referințele la entitatea parametru sunt întotdeauna expandate imediat ce au fost recunoscute, iar după expandarea totală declarația DTD trebuie să fie în conformitate cu regulile gramaticale semnificative.

În cazul documentelor XML cu format corespunzător, entitățile parametru nu trebuie să conțină referințe directe sau indirecte la ele însele.

Entități generale (interne)

Dacă definiția unei entități este doar o valoare, aceasta este denumită entitate generală (sau entitate internă), iar textul cu care se înlocuiește este dat de declarația entității, după cum se arată în exemplul următor:

```
<!ENTITY Shortcut "Acesta este un bloc text pe care il voi folosi foarte des, dar
doresc sa-l tastez numai o data">
```

Aceste entități pot fi utilizate în documentele XML; se apelează astfel:

```
&Shortcut;
```

Entități externe

Dacă o entitate nu este internă, atunci ea este externă. Entitățile externe pot fi folosite pentru a prelua texte din alte fișiere, ca în acest exemplu:

```
<!ENTITY commontext SYSTEM "http://my.home.com/text/block1.xml">
```

Cuvântul cheie `SYSTEM` identifică adresa URL, care poate fi folosită pentru găsirea entității. Un identificator extern poate conține cuvântul cheie `PUBLIC` și, de asemenea, un identificator public. Procesorul XML ar putea încerca să genereze o adresă URL alternativă din acesta, dar dacă nu reușește, trebuie să folosească adresa URL specificată de cuvântul cheie `SYSTEM`.

Fiecare entitate text externă dintr-un document XML poate folosi o codificare diferită pentru caracterele sale. Limbajul XML furnizează o instrucțiune de procesare a declarației de codare, care dacă există, trebuie să fie la începutul unei entități sistem, înaintea oricărei date caracter sau marcaj. În entitatea documentului, declarația de codare este parte a declarației XML; în alte entități, este parte a instrucțiunii procesului de codare.

Pentru validarea unui document XML, procesorul XML trebuie să includă conținutul unei entități text externe. Dacă nu trebuie făcută validarea documentului XML, procesorul XML nu trebuie să includă conținutul unei entități text externe.

Entitățile externe sunt utilizate și pentru a include obiectele nontext, cum sunt de exemplu fișierele grafice. Entitățile nontext externe trebuie să conțină o declarație de notație (`NDATA`), ca în exemplul următor:

```
<!ENTITY myface SYSTEM "../images/me.gif" NDATA GIF>
```

Notațiile identifică după nume formatul entităților binare externe sau aplicația care poate lucra cu formatul. Declarațiile de notație oferă un nume pentru notație și un identificator extern care permite procesorului XML sau aplicației client (cum sunt aplicațiile plug-in pentru browser) să găsească o aplicație ajutătoare care să poată prelucra datele din notație. Următorul exemplu ilustrează declarația a două fișiere grafice ca entități externe și identifică formatul lor ca fiind PCX:

```
<!DOCTYPE BOOK SYSTEM "BOOK,DTD" [
<!-- Figuri pentru acest capitol:-->
<!ENTITY chap1.fig1 SYSTEM "0101.PCX" NDATA PCX>
<!ENTITY chap1.fig2 SYSTEM "0132.PCX" NDATA PCK>
]>
```

Posibilitatea de a declara în mod explicit notația pentru un fișier vă oferă mai multă libertate asupra formatului față de cea pe care o aveți în limbajul HTML. Formatul nu mai este legat de un anume tip de fișier (extensie), ci depinde numai de ceea ce ați declarat că este.

Într-un document XML valid, numele notației folosite în declarația entității *trebuie* să fie declarat.

Entități predefinite

După cum am menționat anterior, caracterul ampersand (&) și paranteza unghi stânga (<), care sunt interpretate în mod normal drept marcaj, trebuie să fie precedate de secvențe escape care folosesc referințe de tip numeric la caracter sau șirurile `&` și `<`.

În plus, paranteza unghi dreapta (>) poate fi reprezentată prin șirul `>`, apostroful (') poate fi reprezentat ca `'`, iar glilimelele (") pot fi reprezentate ca `"`.

Documentele cu format corespunzător nu trebuie să declare entitățile `amp`, `lt`, `gt`, `apos` și `quot`, dar documentele XML valide trebuie să le declare înainte de a le folosi. Toate procesoarele XML trebuie să recunoască aceste entități, indiferent dacă sunt declarate sau nu.

Secțiuni condiționale

Entitățile parametru pot fi utilizate în declarațiile DTD și pentru a transforma secțiunile de marcare pe care le conțin în *secțiuni condiționale*. Aceste secțiuni condiționale pot fi folosite pentru a permite „personalizarea" declarațiilor DTD pentru anumite documente. Personalizarea e realizată prin introducerea declarațiilor entității parametru în subsetul intern al declarației DTD corespunzătoare documentului, astfel încât acestea pot fi folosite drept „comutatoare":

```
<?XML VERSION "1.0" RMD="ALL"?>
<!DOCTYPE BOOK SYSTEM "strict.dtd" [
<!ENTITY %DRAFT "INCLUDE">
<!ENTITY % FINAL "IGNORE">
<![%DRAFT;[
<!ELEMENT BOOK (AUTHOR_COMMENTS*, FRONT, BODY, BACK)>
[[>
<!%FINAL;[
<!ELEMENT BOOK (FRONT, COPYRIGHT, BODY, BACK, SOFTWARE)>
]]>
]>
<BOOK>...
```

În acest document declarația stabilește valoarea entității parametru `DRAFT` ca fiind „`INCLUDE`", iar valoarea parametrului `FINAL` ca fiind „`IGNORE`". Versiunea finală a documentului va schimba valorile celor două entități parametru.

Secțiunile CDATA

Secțiunile `CDATA` (secțiuni de date caracter) vă permit să introduceți cantități mari de caractere de marcare fără a fi nevoie să folosiți secvența escape pentru fiecare caracter & sau < și să ascundeți blocurile de caractere de procesorul XML. Secțiunile `CDATA` pot fi oriunde pot apărea acele date caracter. Ele încep cu secvența `<![CDATA[` și se termină cu secvența `]]>`, după cum se vede în exemplul următor:

```
<![CDATA[ <P>Acest text nu va fi 'vazut' sau procesat.</P>]]>
```

În cazul special al secțiunilor `CDATA`, *datele caracter* reprezintă orice șir de caractere care nu include secvența `]]>`, care încheie secțiunea `CDATA`.

Secțiunile `CDATA` nu pot fi imbricate.

Legăturile XML

Unul dintre aspectele limbajului XML pe care l-am ocolit până acum în mod intenționat este cel referitor la legături. XML furnizează legături mult mai puternice față de SGML și HTML și aceste legături merită să fie tratate separat pentru a le judeca la adevărata valoare.

Concepute în anii de cercetare în domeniul legăturilor, mecanismele de legături ale XML (cunoscute sub numele de XLL, limbajul de legare XML), sunt bazate pe idei împrumutate din HTML, HyTime (Hypermedia/Time-Based Structuring Language, ISO 10744:1992) și o aplicație a limbajului SGML, larg folosită în domeniul academic, denumită Inițiativa de codare a textului (Text Encoding Initiative–TEI).

Legăturile XML sunt interoperabile cu limbajul HTML. Ele folosesc același atribut `HREF` și dau valorii acestuia aceeași semnificație ca în HTML. Valoarea unui atribut `HREF` este adresa URL care identifică *resursa*, ținta legăturii, calificată opțional de # și un *identificator de fragment* (sau ?) și o întrebare. Legăturile XML sunt și ele conforme cu convențiile 1738, care stabilesc caracterele care pot apărea în URL.

Totuși, legăturile XML adaugă legăturilor `HREF` următoarele caracteristici:

- Permit elementelor de absolut orice tip să fie sursa unei legături
- Definesc proprietăți și comportamente suplimentare pentru toate legăturile
- Introduc conceptul de legături out-of-line (în afara documentului sursă)
- Definesc semnificația precisă a identificatorului de fragment când ținta legăturii este un document XML.

Tipurile de legături

Limbajul XML permite elementelor de oricare tip să funcționeze ca un element de legătură, prin specificarea atributului rezervat `XML-LINK`, pentru fiecare element care va funcționa ca element de legătură. În plus, valoarea `XML-LINK` a atributului indică ce tip de legătură exprimă elementul, după cum arată exemplul de mai jos:

```
<A XML-LINK="SIMPLE" HREF="http://www.home.com/">Our home Page</A>
```

Această declarație indică procesorului XML că procesează o legătură simplă (inline).

Atributul `XML-LINK` poate fi tipul de legătură `SIMPLE` (simplă), `EXTENDED` (extinsă), `LOCATOR` (locator), `GROUP` (grup) sau `DOCUMENT` (document). Două din aceste tipuri sunt descrise în secțiunile următoare.

Legături simple

Legătura *simplă* poate conține numai un locator, iar atributele adresei sunt atașate în mod direct elementului de legătură. Legăturile simple sunt de obicei inline și întotdeauna unidirecționale.

Următorul exemplu prezintă declarația pentru o legătură simplă, care imită comportarea unui tip de legături cu care probabil sunteți obișnuit în documentele HTML; noul conținut îl înlocuiește pe cel vechi în momentul în care faceți clic pe legătură:

```
<!ELEMENT SIMPLE ANY>
<!ATTLIST EXTENDED
    XML-LINK        CDATA                   #FIXED "SIMPLE"
    ROLE            CDATA                   #IMPLIED
    TITLE           CDATA                   #IMPLIED
    INLINE          (TRUE|FALSE)            "TRUE"
    CONTENT-ROLE    CDATA                   #IMPLIED
    CONTENT-TITLE   CDATA                   #IMPLIED
    SHOW            (EMBED|REPLACE|NEW)     "REPLACE"
    ACTUATE         (AUTO|USER)             "USER"
    BEHAVIOR        CDATA                   #IMPLIED>
```

Legăturile simple sunt foarte asemănătoare elementelor `A` din limbajul HTML.

Legăturile extinse

Legătura *extinsă* poate implica oricâte resurse, care nu este nevoie să se afle în același document fizic cu celelalte. Locatorii legăturii extinse sunt păstrați în elementele pe care aceasta le conține, fiecare putând avea propriul set de atribute. Se presupune că o aplicație (cum este de exemplu browserul Web) va urma toate legăturile din legătura extinsă, după cum se vede în exemplul de mai jos:

```
<!ELEMENT EXTENDED ANY>
<!ELEMENT LOCATOR ANY>
<!ATTLIST EXTENDED
    XML-LINK            CDATA                   #FIXED "EXTENDED"
    ROLE                CDATA                   #IMPLIED
    TITLE               CDATA                   #IMPLIED
    INLINE              (TRUE|FALSE)            "TRUE"
    CONTENT-ROLE        CDATA                   #IMPLIED
    CONTENT-TITLE       CDATA                   #IMPLIED
    SHOW                (EMBED|REPLACE|NEW)     "REPLACE"
    ACTUATE             (AUTO|USER)             "USER"
    BEHAVIOR            CDATA                   #IMPLIED>
<!ATTLIST LOCATOR
    XML-LINK            CDATA                   #FIXED "LOCATOR"
    ROLE                CDATA                   #IMPLIED
    HREF                CDATA                   #REQUIRED
    TITLE               CDATA                   #IMPLIED
    SHOW                (EMBED|REPLACE|NEW)     "REPLACE"
    ACTUATE             (AUTO|USER)             "USER"
    BEHAVIOR            CDATA                   #IMPLIED>
```

Legăturile extinse pot să fie inline (în documentul sursă) sau out-of-line (în afara documentului sursă) și pot fi utilizate pentru legături multidirecționale, legături cu date numai pentru citire (read-only) ș.a.

Atributele elementului cu legături extinse precizează modul de comportare al elementelor interioare, astfel încât să poată primi atributele `ROLE` și `TITLE` pentru a putea descrie comportarea lor.

Rețineți că multe atribute pot fi plasate atât în elementele de legătură părinte, cât și în elementele privind adresa pe care o conțin. Dacă un astfel de atribut este precizat în elementul de legătură, dar nu și în elementul locator, valoarea elementului de legătură e folosită pentru procesarea elementului locator.

Legături inline și out-of-line

O legătură este inline sau out-of-line după cum este valoarea atributului `INLINE`. Valoarea `TRUE`, care e prestabilită, are semnificația că întregul conținut al elementului de legătură este considerat parte integrantă a legăturii, cu excepția oricărui element copil tip locator (care este considerat parte a mecanismului elementului de legătură).

Când legătura este inline, pot fi folosite atributele `CONTENT-ROLE` și `CONTENT-TITLE` pentru furnizarea informațiilor privind titlul și rolul pentru această resursă de „conținut". Dacă `INLINE` este `FALSE`, atributele `CONTENT-ROLE` și `CONTENT-TITLE` nu au efect.

Informații asociate cu legăturile

Ca și limbajul HTML, XML permite asocierea informațiilor cu legăturile. Totuși, XML definește modul de folosire al aspectelor principale conținute în aceste informații, ceea ce vă dă posibilitatea să controlați mai bine modul de comportare al legăturii.

Informația e asociată cu legăturile prin următoarele atribute:

- **`ROLE`:** Acesta este un șir care identifică semnificația legăturii în aplicație. Fiecărui participant la legătură i se acordă propriul rol. Nu există valori predefinite pentru rolul legăturilor; rolurile atribuite și semantica acestora sunt specifice aplicației XML în care sunt folosite.
- **`HREF`:** Identifică ținta.
- **`TITLE`:** Stabilește o etichetă pentru legătură. Aplicația XML decide dacă utilizează această etichetă și la ce să o folosească.
- **`SHOW` și `ACTUATE`:** Precizează cum și când e urmată legătura.
- **`INLINE`:** Precizează dacă elementul legăturii este inline.

Comportarea legăturii

Momentul și efectele urmării (realizării) unei legături sunt exprimate de atributele `SHOW` și `ACTUATE`.

SHOW

Atributul `SHOW` precizează modul cum ar trebui afișată sau procesată ținta; are următoarele valori:

- **`EMBED`:** Prin urmarea legăturii, ținta trebuie să fie înglobată în corpul sursei de unde a pornit legătura.
- **`REPLACE`:** Prin urmarea legăturii, ținta trebuie să înlocuiască sursa de unde a plecat legătura.
- **`NEW`:** Prin urmarea legăturii, ținta trebuie să fie afișată sau procesată într-un context nou (ca de exemplu într-o fereastră nouă a browserului), fără să afecteze contextul sursei de unde a plecat legătura.

Comportarea legăturilor XML este conformă celei precizate de atributul `EMBED` pentru elementele `IMG` și de atributul `REPLACE` pentru elementele `A` în limbajul HTML.

ACTUATE

Atributul `ACTUATE` precizează când trebuie urmată o legătură; are următoarele două valori:

- **`AUTO`:** Când e întâlnită această valoare, afișarea sau prelucrarea sursei nu este considerată completă până ce nu a fost rezolvată legătura.
- **`USER`:** Legătura nu este urmată până ce nu apare o cerere externă explicită pentru acest lucru.

Comportarea legăturilor XML este conformă celei precizate de atributul `AUTO` pentru elementele `IMG` și de atributul `USER` pentru elementele `A` în HTML.

Adresarea

Atributele `HREF` ale limbajului XML sunt interoperabile cu cele din limbajul HTML. Totuși, XML oferă posibilități mai sofisticate de adresare a resursei țintă când este vorba despre un document XML. Puteți folosi acest mod de adresare special în cadrul paginilor HTML pentru a adresa părți ale documentelor XML.

Când un locator identifică un document XML ca țintă, valoarea locatorului poate conține o adresă URL, un identificator de fragment sau amândouă informațiile.

Dacă există, adresa URL este tratată ca o adresă URL standard și identifică *resursa conținut* care este ținta legăturii (documentul). Dacă nu există nici o adresă URL, se presupune că resursa conținut este documentul în care e găsită legătura.

Orice *identificator de fragment* e tratat ca un *poiter extins*. Dacă este furnizat un pointer extins, resursa țintă este o subresursă a resursei conținut. Un identificator de fragment este precedat de un *conector* care identifică cum și unde trebuie localizat și procesat fragmentul:

- **#**: Un fragment standard, stil HTML; întreaga resursă conținut este transmisă clientului, care apoi trebuie să găsească fragmentul.
- **?XML-PTR=**: Interogare XML; serverul interpretează identificatorul de fragment (pointerul extins) și transmite clientului *numai* fragmentul cerut.
- **|**: Conector specific limbajului XML; oricare din metodele precedente poate fi folosită pentru a aduce resursa cerută.

Pointerii extinși

Pornind de la elementul rădăcină (elementul nominalizat în declarația DTD), un element care conține alte elemente este considerat „părintele" acestora, iar acestea sunt, la rândul lor, „copiii" primului. Prin extinderea acestor relații părinte-copii, puteți face diagrama arborelui structurii elementelor unui document XML (în mod normal, nu puteți face această diagramă a structurii elementelor în HTML, deoarece nimic nu vă împiedică să folosiți, de exemplu, elementele H1, H2, H3 și H4 în orice ordine doriți, chiar dacă într-un document HTML „structurat" ar trebui să le folosiți în mod ierarhic).

Pointerii extinși lucrează cu arborele elementelor definit de elementele din documentul XML. Ele descriu proprietățile elementelor din document, cum ar fi tipul elementelor, sau valorile atributelor sau fac numai număratoarea elementelor.

Forma de bază a unui pointer extins este o serie de termeni ai locației, fiecare precizând o locație absolută sau una relativă

Termeni de locație absoluți

În cazul unei locații absolute, fiecare termen al locației lucrează în contextul unei surse de locație. În mod prestabilit, sursa locației pentru primul termen al locației este elementul rădăcină al documentului XML. Fiecare pointer extins poate începe de la unul dintre următorii termeni ai locației:

- **ROOT**: Cuvântul cheie ROOT precizează faptul că sursa locației este elementul rădăcină al sursei. Aceasta este comportarea predefinită.
- **HERE**: Cuvântul cheie HERE specifică faptul că sursa locației pentru primul termen al locației din serie este elementul de legătură care conține locația și nu elementul rădăcină prestabilit.
- **DITTO**: Cuvântul cheie DITTO precizează faptul că sursa locației pentru primul termen al locației este sursa locației specificată de primul pointer complet.
- **ID**: Cuvântul cheie ID precizează faptul că sursa locației pentru primul termen al locației este elementul din sursă care are atributul de tip ID cu o valoare care coincide cu numele dat (de exemplu, ID(*Name*)).
- **HTML**: Cuvântul cheie HTML selectează primul element al cărui tip este A și care are un atribut NAME a cărui valoare este egală cu valoarea NAME furnizată (este analog cu # din documentele HTML).

Termenii locației relative

Termenul locației relative este un cuvânt cheie, urmat de unul sau mai mulți pași. Cuvintele cheie permise sunt:

- **CHILD**: Selectează elementele copil din sursa locației.
- **DESCENDANT**: Selectează elementele care apar în conținutul sursei locației (de exemplu, arborele elementelor care încep cu acest element).

- **ANCESTOR**: Selectează elementele în care se găsește sursa locației (de exemplu, arborele elementelor care începe cu părintele elementului curent).
- **PRECEDING**: Selectează elementele care apar înainte de sursa locației.
- **PSIBLING**: Selectează elementele frate care preced sursa locației (de exemplu, ceilalți copii ai elementului părinte care apar înaintea acestui element).
- **FOLLOWING**: Selectează elementele care apar după sursa locației.
- **FSIBLING**: Selectează elementele frate care urmează sursa locației (de exemplu, alt copil al elementului părinte care urmează după acest element).

Selectarea țintelor

Elementele care coincid cu referințele locațiilor sunt denumite *candidați*. Candidații pot fi selectați după numărul de apariții, tipul elementelor și numele și valoarea atributului.

Un cuvânt cheie limitează candidații posibili pentru o referință a locației la acele elemente care posedă proprietatea cerută, corespunzătoare sursei locației. Cuvântul cheie e urmat de unul sau mai mulți pași. Fiecare pas definește o instanță, un tip de element opțional și un tip și o valoare pentru atributul opțional.

Instanța numără candidații. Poate fi un număr pozitiv (numărând înainte de la primul candidat), un număr negativ (numărând înapoi de la ultimul candidat) sau valoarea specială ALL, pentru a selecta toți candidații:

```
(4) – selecteaza locatia celui de-al patrulea candidat
(-2) – selecteaza locatia celui de-al doilea candidat din coada
```

Instanța poate fi urmată de virgulă și de tipul elementului, care e opțional, dar recomandat și care poate lua următoarele valori:

- **Name**: Selectează elementele de tipul Name.
- *****: Selectează orice tip de element.
- ***CDATA**: Selectează „pseudoelementele" care conțin numai text.

Iată câteva exemple de folosire a acestora:

- (3,P) Selectează al treilea element <P>.
- (-1,*) Selectează ultimul element.
- (2,*CDATA) Selectează al doilea interval de text fără etichetă.

Dacă se specifică tipul elementului, acesta poate fi precizat cu ajutorul numelui și valorii atributului. Numele atributului poate lua următoarele două valori:

- *****: Corespunde oricărui nume de atribut.
- **Name**: Precizează tipul atributului prin Name.

Valoarea atributului poate fi:

- ***IMPLIED**: Corespunde atributelor pentru care nu a fost specificată nici o valoare și nu există valoare prestabilită.
- *****: Corespunde oricărei valori.
- **Name**: Corespunde valorii Name.
- **"value"** : Corespunde valorii dintre glilimele.

Următoarea linie selectează primul element copil al sursei locației care este un element <P>, pentru care atributul FONT a fost lăsat neprecizat:

```
CHILD(1,P,FONT,*IMPLIED)
```

Domeniile pointerilor extinși

Un identificator de fragment poate conține numai un pointer extins sau doi pointeri extinși, delimitați prin șirul . . , pentru a defini un interval al textului.

Fragmentul e presupus ca fiind totul de la începutul țintei primului pointer extins (*sursa locației*) și până la sfârșitul celui de-al doilea. Puteți selecta un interval de elemente ca țintă a legăturii, în locul întregului document, cum făceați în limbajul HTML.

Următorul identificator de fragment selectează textul de la începutul primului element copil din sursa locației care este un element <P> și se termină la sfârșitul ultimului copil care este element <P>:

```
(CHILD(1,P) .. (CHILD(-1,P)
```

Grupuri de legături extinse

Elementele grupurilor de legături extinse sunt folosite pentru păstrarea listelor de legături la alte documente, care formează un grup de documente interlegate.

Fiecare element este identificat cu atributul HREF al elementului documentului cu legăturile extinse care este un element copil al grupului GROUP. Valoarea atributului HREF este o locație.

Aceste elemente sunt recunoscute folosind atributul XML-LINK cu valoarea GROUP sau DOCUMENT.

Elementul GROUP conține unul sau mai multe elemente DOCUMENT, fiecare indicând o resursă (sau o subresursă, cu sintaxa pointerului extins) care constituie o parte a grupului de documente.

De exemplu, un document XML poate conține următoarele:

```
<GROUP>
<DOCUMENT HREF="http://www.home.com/bookmarks1.xml"/>
<DOCUMENT HREF="http://www.home.com/bookmarks2.xml"/>
</GROUP>
```

În această declarație, GROUP și ELEMENT sunt identificate ca fiind elemente de legătură, astfel încât vor fi procesate documentele `bookmarks1.xml` și `bookmarks2.xml` pentru a căuta legăturile care indică acest document.

Atribute XML-LINK cu valoare fixă

Pentru a evita declararea atributului special XML-LINK de fiecare dată când folosiți un element, puteți să-l declarați o dată în declarația DTD ca valoare fixă:

```
<!ATTLIST MYLINK
        XML-LINK CDATA #FIXED "SIMPLE">
```

Stilul XML

În principiu, foile de stil de nivel 1 (CSS1) pot fi folosite pentru a da un format simplu documentelor XML. Documentele XML, ca și cele HTML, conțin elemente cu nume, identități ș.a.m.d. – aceleași blocuri de date text pe care foile CSS1 le folosesc pentru a defini regulile formatului documentelor HTML.

Totuși, foile CSS1 sunt special concepute pentru HTML și unele din regulile acestora se bazează pe practica specifică a codării HTML. De exemplu, pseudoclasele, cum sunt `A:link`, `A:visited` și `A:active`, permit legăturilor să arate diferit când sunt selectate sau după ce au fost urmate. Pseudoelementele, cum sunt de exemplu `:first-line` și `:first-letter`, și clasele normale folosite de foile CSS1 ar trebui de asemenea să fie adăugate ca atribute ale elementelor din declarațiile DTD ale documentelor XML, pentru a permite ca valorile lor să fie folosite în foile de stil CSS1 pentru documentele XML.

Limbajul XML are propriul limbaj de stil, numit XSL, care este mult mai puternic decât CSS1; va rămâne de văzut dacă limbajul XSL va fi folosit împreună cu foile CSS1 sau în locul acestora.

NOTĂ

Limbajul XSL este încă în stadiu incipient de dezvoltare; în momentul în care se scriu aceste rânduri, standardul este disponibil numai ca o propunere preliminară.

XSL este bazat pe o formă a limbajului DSSSL, numită DSSSL-o (o de la *online*). DSSSL (Limbaj de specificare a semanticii și a stilului documentelor: Standardul ISO/IEC 10179:1996) este un limbaj care se bazează pe Scheme, un dialect al limbajului de programare LISP, care suportă transformarea, interogarea și prelucrarea documentelor SGML. Limbajul de transformare al DSSSL nu este folosit în XSL. În locul acestuia, XSL folosește forma oficială a limbajului Java Script, așa cum a fost definită de ECMA (Asociația Europeană pentru Standardizarea Informației și Sisteme de Comunicații) în Standardul 262 „ECMAScript". Partea limbajului DSSSL referitoare la foi de stil a fost de asemenea redusă la un minim esențial, necesar pentru a ajuta tipărirea paginilor Web.

Cum lucrează XSL

În esență, XSL este un mecanism condus prin date. Când un document XML este formatat (fie pentru tipărire pe hârtie, fie pentru afișare pe ecran), sunt citite una sau mai multe foi de stil XSL. Foile de stil XSL pot fi precizate chiar în documentul XML sau le puteți selecta.

Procesarea documentelor XML e determinată de scanarea structurii sale și fuzionarea cu *specificația de formatare* obținută din foile de stil active. Aceste instrucțiuni sunt apoi folosite pentru a a crea *obiecte de curs* (flow objects), precum paragrafele sau tabelele, care determină codarea aplicată documentului. Procesul de fuziune produce o structură arbore a cursului obiectelor, numită *arborele obiectelor de curs*.

Declarația de bază în XSL este *regula de construire*, care declară ce trebuie făcut cu un element. Mai precis, aceasta precizează cum vor fi create obiectele de curs și ce caracteristici vor avea fiecare. De exemplu, regula de construcție pentru un element P ar trebui să specifice că un obiect de curs paragraf trebuie adăugat la arborele obiectelor de curs, cu următoarele caracteristici:

```
font-size: 12pt
first-line-start-indent: 18pt
quadding: left
```

Această declarație stabilește caracterele din paragraf la 12 puncte, cu o indentare de 18 puncte a primei linii și o aliniere la stânga a textului, aplicată întregului paragraf.

Procesarea în limbajul XSL

Deși XSL reprezintă o personalizare a DSSSL, care a fost redus la strictul necesar, asemănător modului în care XML e considerat o personalizare a SGML, adoptarea sa de către ECMAScript vă oferă accesul la cele mai multe caracteristici ale unui limbaj de programare complet, necesare pentru procesarea documentelor XML. ECMAScript poate fi folosit pentru a rezolva calcule, testarea diferitelor condiții ș.a. Poate fi de asemenea folosit pentru a realiza instrucțiuni complexe pentru procesarea elementelor particulare sau chiar a caracterelor dintr-un document XML.

Aproape oricare aspect al structurii documentelor XML poate fi utilizat pentru controlul procesării acestuia. Cea mai mare parte a regulilor de construcție operează la nivel

element, dar puteți ușor îmbunătăți comportarea acestora prin testarea proprietăților unui element, precum atributele sale și valorile acestora sau elementele strămoși.

Deși în mod prestabilit, întregul document va fi procesat în ordinea originală, XSL vă oferă accesul la întreaga structură a elementelor în oricare punct din document. Acest lucru înseamnă că puteți include părți ale documentului astfel încât, spre exemplu, să realizați o tablă de materii din anteturile capitolului, pe care s-o poziționați la începutul documentului. Este de asemenea ușor să anulați părți ale unui document care nu doriți să fie afișate în acest context, ceea ce permite caracteristici precum conținutul condițional, conținutul dinamic și conținutul cu acces controlat.

Viitorul limbajului XML

Până acum, această discuție privind limbajul XML a fost destul de tehnică, așa că ar fi normal să ne tragem puțin sufletul, prin recapitularea rapidă a unora dintre aplicațiile pentru care XML este deja folosit:

- Channel Definition Format (CDF) este o aplicație XML aflată în spatele *tehnologiei push*. (Pentru mai multe detalii referitoare la CDF, vezi Capitolul 39, „Să înțelegem limbajul XSL". În loc să navigați de la un sit Web la altul și de la o pagină la alta în sit, obțineți un „abonament" la un anumit furnizor de conținut. În loc să *atrageți* paginile Web către dumneavoastră, permiteți altcuiva să *împingă (push)* aceste pagini către dumneavoastră, când stabilește acest lucru. După ce v-ați exprimat opțiunile de abonare, vă puteți retrage și savura informația difuzată prin Web (împinsă către ecranul dumneavoastră), din postura de spectator, aproape la fel cum ați privi la televizor.
- Open Software Description (OSD) este o aplicație XML care descrie componentele software, versiunile și structura acestora, precum și relația cu alte componente. Combinat cu CDF, OSD permite instalarea și modernizarea automată a unor aplicații software din Internet.
- Sun's Answerbook este o aplicație pentru distribuția documentației tehnice. Folosind limbajul XML ca format de stocare, manualele tehnice sunt convertite în HTML pentru publicarea pe Web.
- Chemical Markup Language (CML) este o aplicație XML care, folosită cu miniaplicația Jumbo Java, permite descrierea și vizualizarea formulelor chimice și a modelelor moleculare cu un browser Web.
- Mathematics Markup Language (MathML) este o aplicație XML pentru marcarea elementelor matematice. Este o aplicație interesantă, deoarece folosește structuri de marcare paralele, atât pentru reprezentările, cât și pentru structurile semantice ale matematicii.
- OpenTag este o aplicație XML pentru marcarea fragmentelor de text pentru traduceri și localizare, indiferent de software-ul utilizat pentru crearea textului. Această aplicație reprezintă un exemplu care ilustrează modul cum XML poate crea aplicații utile care au chiar mai puține elemente decât limbajul HTML.
- Handheld Device Markup Language (HDML) este o aplicație XML destinată afișării informațiilor (inclusiv a paginilor Web) pe dispozitive electronice de buzunar, cum ar fi Asistenții digitali personali (Personal Digital Assistants) și telefoanele mobile echipate cu ecrane de afișare.

Această listă de aplicații XML nu este deloc completă; există multe altele, începând cu cele simple (Frequently Asked Question Markup Language – FAQML) și până la cele extrem de complexe (inițiativa de interschimbare a datelor electronice bazată pe limbajul XML – XML/EDI).

Chiar dacă specificațiile limbajului XML nu au fost puse într-o formă definitivă, aplicațiile pentru care este folosit ilustrează un palmares impresionant. În mai puțin de un an, XML a evoluat de la stadiul de idee la cel de limbaj de succes. Indiferent de forma finală a specificației, nu există nici o îndoială asupra importanței viitoare a limbajului XML, nu numai în tehnologia Web, dar și în toate formele de documentație electronică.

Rezumat

În acest capitol ați studiat despre:

- Scopul și natura limbajului XML
- Regulile sintactice și structurale de bază ale limbajului XML
- Noțiuni de bază privind declarațiile tipului documentului (DTD)
- Probleme esențiale privind mecanismul de legare al limbajului XML (XLL)
- Limbajul de stil al XML (XSL)
- Cum și unde este folosit limbajul XML

Acest capitol a insistat pe descrierea XML (și XSL) ca un limbaj, așa că a fost într-o oarecare măsură tehnic. În următoarele capitole, voi compara XML cu HTML pe principii mai mult funcționale și apoi voi merge mai departe și voi traduce aceste descrieri tehnice în practică prin discuții mai concrete.

Capitolul 37
Comparație între limbajele XML și HTML

de Simon North

În acest capitol:

- **HTML mai are un cuvânt de spus! 582**
- **Documente inteligente 583**
- **Validarea 583**
- **Legături mai puternice 585**
- **Stiluri inteligente 587**
- **Nu poate Java, JavaScript sau ActiveX să rezolve problema? 588**

În capitolul 36, „Ce este limbajul XML?", v-am oferit o descriere destul de tehnică a limbajului XML (Extensible Markup Language). În acest capitol, veți utiliza cunoștințele acumulate pentru a privi mai îndeaproape diferențele dintre XML și HTML.

Însă încercarea de a compara XML cu XTML este ca și cum ai încerca să compari Departamentul Trezoreriei S.U.A. cu bancnota de 100 $. HTML (bancnota de 100 $) este un limbaj de marcare, reprezentând doar o aplicație a SGML. El este constituit dintr-un set fix de elemente (etichete) și un set bine definit de relații între acestea. XML (Trezoreria S.U.A.) reprezintă un dialect al SGML potrivit pentru utilizarea în Internet. XML reprezintă un fel de trusă de scule care vă permite să creați atâtea limbaje de marcare câte doriți (nu numai bancnote de 100 $, dar și de 1 $, 5 $, 10 $, 50 $ sau de orice valoare).

Ca aplicație a SGML, limbajul HTML este diferit fundamental de o aplicație realizată folosind XML. Deci, pentru a compara XML și HTML, trebuie să observați diferențele dintre o aplicație XML și HTML. Aplicația XML poate fi totuși mult mai sofisticată decât limbajul HTML, cum este de exemplu Mathematical Markup Language (MathML), sau poate fi mult mai simplă decât HTML, cum este Handheld Device Markup Language (HDML). Pentru a face o comparație echitabilă și semnificativă, trebuie să vă concentrați asupra următoarelor caracteristici generale, în locul unora particulare:

- Posibilitățile de extindere
- Marcarea semantică
- Validarea
- Legăturile
- Limbajul stilului

După citirea acestui capitol, ar trebui să aveți o imagine generală a principalelor diferențe dintre XMl și HTML.

HTML mai are un cuvânt de spus!

Înainte de a începe comparația detaliată a limbajelor XML și HTML, trebuie subliniat un lucru, cu riscul de a-l auzi des și la nesfârșit în lunile următoare: *XML nu reprezintă un înlocuitor al HTML.*

HTML a fost conceput ca un limbaj simplu și ușor de implementat. Peste ani a devenit mai complex, dar chiar și în cazul versiunii 4.0 poți să înveți pe cineva setul de etichete de bază

în câteva ore. Folosind unul dintre multele pachete software disponibile, o persoană fără experiență poate crea o pagină de bază simplă în câteva minute.

Limbajul HTML face o treabă minunată pentru paginile Web simple – și aici este diferența principală între XML și HTML. HTML pune accentul pe interfața cu utilizatorul, iar XML acordă atenție datelor. HTML descrie prezentări și interacțiunea utilizatorului; XML descrie informația. Cele două limbaje pot coexista în pace, deoarece îndeplinesc sarcini diferite. Acum, fiind spuse aceste lucruri, parcurgeți subcapitolul următor pentru a vedea cât de bun este XML în comparație cu HTML.

Documente inteligente

HTML este ca briceagul armatei elvețiene pentru celelalte limbaje de marcare. Vă oferă un set de elemente general, multifuncțional pentru afișarea textelor cu un browser Web. Din punct de vedere oficial, conform DTD (Document Type Definition) din HTML, singurul element dintr-un document HTML care trebuie etichetat este elementul `TITLE`, chiar dacă acesta este probabil unul din elementele lăsate cel mai adesea pe dinafară. Restul etichetelor elementelor sunt opționale (sau, mai precis, pot fi deduse din context) și sunt puține sau inexistente reguli obligatorii care să spună ce etichete trebuie folosite și unde. Un lucru este sigur totuși; puteți folosi numai etichete pentru elemente definite.

XML nu este denumit pe degeaba Limbaj de marcare extensibil. Acesta este, după cum sugerează și numele, extensibil și aceasta este una din cele mai importante diferențe dintre XML și HTML. HTML posedă un set limitat de elemente, bine definit. În XML, aveți libetate totală să definiți orice elemente doriți. Puteți, dacă doriți, să folosiți pur și simplu elementele HTML existente și să adăugați câteva ale dumneavoastră; ar trebui, bineînțeles, să furnizați o versiune compatibilă cu XML a definiției DTD pe care doriți să o utilizați. Pe de altă parte, puteți folosi un set complet diferit de elemente pentru a satisface necesitățile aplicației dumneavoastră.

Din punct de vedere al posibilităților există două tipuri de tehnici de marcare. Una este orientată spre prezentare, definind paragrafele, liniile, caracteristicile textului ș.a.m.d. Cealaltă este orientată spre conținut și ignoră complet metoda de prezentare, preferând să identifice conținutul informației. Acest tip de tehnică de marcare folosește obiecte precum avertizările, procedurile, introducerile, capitolele ș.a.m.d. Extensibilitatea limbajului XML vă permite să introduceți sensul semantic în documentele dumneavoastră, transformându-le în „documente inteligente", care pot descrie conținutul propriu într-o manieră semnificativă. Puteți, spre exemplu, să precizați dacă o listă este de fapt procedură de reparare sau să indicați că paragraful scris cu text de culoare roșie este de fapt un mesaj de avertizare.

Probabil că oricare doi autori de documente HTML nu s-ar pune de acord să folosească `H5` în loc de `H6` și nu există nimic care să vă împiedice să le folosiți în orice ordine vă place. Însă în XML le puteți identifica drept titluri de secțiuni, titluri de capitole ș.a.m.d.

Validarea

Deși există Definiția tipului documentelor HTML (DTD) și pachete de software care fac verificarea documentului dumneavoastră HTML pentru a vedea dacă este valid, validarea e mai mult excepție decât regulă. Dintre toate paginile actuale din Web, mai puțin de un procent sunt recunoscute a fi într-adevăr valide HTML.

Unele pachete de creație, cum ar fi HoTMetaL de la SoftQuad, încearcă să vă pună pe drumul cel bun al realizării documentelor HTML valide. Totuși, pentru fiecare dintre

aceste pachete sunt alte câteva sute care, sub masca ofertei unui mediu aproape WYSIWYG (precum editorul furnizat de Netscape ca parte a setului complet Communicator), produc echivalentul HTML al codului spaghetti. Dacă aspectul final este acceptabil, nu contează tipul monstruos de cod HTML care a fost realizat; în final, cui îi pasă cum arată codul?

Pentru această situație, vina nu le aparține totuși integral producătorilor de software. Una dintre primele reguli nescrise ale sitului Web este aceea care spune că pachetele de creație trebuie să încerce să respecte în mod strict limbajul HTML, dar browserele ar trebui să fie libere să accepte orice și să încerce să înțeleagă cât de mult posibil. Din nefericire, unii interpretează această libertate într-un mod cam prea larg.

În primul rând, producătorii de browsere au început să adauge propriile elemente HTML, apoi diverși autori individuali au început să forțeze regulile, experimentând pentru a vedea dacă pot introduce unele efecte interesante, iar unii furnizori de servicii Internet au început să introducă elemente locale, pe care serverele Web individuale le puteau înțelege. Aceasta nu înseamnă o condamnare a eforturilor depuse de aceste persoane. Puteți obține un efect de defilare interesant în Netscape folosind 40 sau 50 elemente `TITLE` la începutul unui document HTML și declarația `<META REFRESH=`, utilizată foarte eficient pentru redirectarea automată a utilizatorilor browserului Netscape. A condamna aceste „inovații", ar însemna să neg natura umană, încercând să limitez inventivitatea și bogăția de resurse care a făcut din Web ceea ce este.

Totuși, Webul nu mai este ceea ce a fost cu câțiva ani în urmă. Este evident că Webul a devenit o platformă comercială. Este larg acceptat, iar persoanele care se gândesc să publice materiale în Web au preocupări foarte îndeparte de cele ale studenților, care nu doreau decât ceva care să arate frumos. Acești utilizatori din „clasa industrială" vor garanții că vor avea un cod care să arate și să se comporte la fel pe toate platformele (MacOS, Windows, IRIX, Solaris; diferite sisteme de operare, diferite rezoluții ale ecranului – lista combinațiilor posibile e aproape fără sfârșit). Utilizatorii doresc, de asemenea, ca aceste garanții să fie durabile și, dacă e posibil, gata de a fi implementate. Pe scurt, acești utilizatori doresc o cale de verificare a corectitudinii codului HTML; ei doresc mecanisme formale de validare, iar HTML nu le poate oferi.

Validarea documentelor HTML necesită urmărirea de către dumneavoastră a unei anumite definiții DTD asociate cu un anume browser Web sau o anumită versiune a HTML sau necesită aplicarea unui pachet de validare separat. Capacitatea de validare a documentelor HTML este o soluție reparatoare. Spre deosebire de aceasta, capacitatea de validare a documentelor XML este o caracteristică intrinsecă a limbajului. Dumneavoastră nu sunteți obligat să validați propriile documente XML, așa cum nu sunteți obligat să validați documentele HTML, dar folosirea declarației DTD în XML face acest lucru mai simplu.

În locul definiției DTD care este „ancorată bine" în browserul Web, precum este definiția DTD din HTML, puteți furniza definiția DTD din XML ca un document extern sau puteți să o înglobați la începutul documentului XML. Dacă doriți, puteți pune o parte comună a DTD, numită *subset extern*, într-un document extern și puteți include o parte locală a DTD, numit *subset intern*, în fiecare document. În capitolul 38, „Utilizarea limbajului XML", veți vedea o metodă de folosire a acestei caracteristici pentru a reduce cantitatea de lucru la crearea paginilor HTML, folosind crearea bibliotecilor comune de elemente grafice.

Pentru a preciza dacă documentele XML create de dumneavoastră necesită validarea, adăugați o declarație inițială la declarația XML (numită *declarația de marcare cerută – RMD*) la începutul documentului XML:


```
<?XML VERSION="1.0" ENCODING="UTF-8" RMD="ALL" ?>
```

Cuvântul cheie `ALL` semnifică faptul că trebuie folosită declarația DTD, cuvântul cheie `INTERNAL` specifică faptul că trebuie utilizat numai subsetul intern, iar cuvântul `NONE` indică faptul că declarația DTD nu e cerută (chiar dacă există).

Mecanismul de validare

Ce este de fapt validarea? Declarația DTD definește structura elementelor care pot fi conținute într-un tip sau clasă de documente. Această structură spune ce elemente există, ce elemente sunt conținute în alte elemente și de câte ori pot apărea ele. Ea identifică elementele obligatorii și opționale și, când proiectantul consideră necesar, definește ordinea prescrisă în care trebuie să apară elementele.

Fiecare element poate avea atribute declarate pentru el, iar aceste atribute sunt declarate cu tipuri specifice de valori. Dacă doriți, puteți declara seturi de valori pentru atribute, din care poate fi aleasă o valoare, sau puteți chiar declara valori predefinite, în cazul în care acestea nu sunt declarate în document.

Folosind declarațiile în definiția DTD, *validarea* este procesul prin care se verifică un anume document (denumit din punct de vedere tehnic *instanță*) pentru a se vedea dacă toate elementele care trebuie să apară există și apar în ordinea cerută. În plus, declarațiile și valorile fiecărui atribut sunt verificate pentru a vedea dacă valorile date sunt permise.

Validarea poate fi un proces riguros, în funcție de natura definiției DTD. HTML este un exemplu de DTD foarte permisiv, deoarece aproape orice lucru este opțional și poate fi omis sau declarat în orice ordine; nimic nu vă oprește, de exemplu, să folosiți o etichetă `H1`, urmată de una `H6` și apoi de `H2`. Dacă definiția DTD este strictă (cum sunt unele DTD folosite pentru aplicațiile militare), conținutul unei instanțe poate fi verificat exact până la cel mai mic detaliu. Este posibil chiar, de exemplu, să se verifice dacă data a fost precizată corect ca an, urmat de numele lunii scris sub formă de text, urmat de zi. În Capitolul 38, veți vedea câteva exemple practice de scriere a declarației DTD în XML, dar important este faptul că dimensiunea controlului pe care-l exercitați asupra conținutului documentului XML este complet în mâinile dumneavoastră, în calitate de autor al declarației DTD.

Legături mai puternice

Codarea a parcurs o cale lungă în ultimii doi ani, de la folosirea editorului de text pentru crearea manuală a codului HTML până la generarea automată a codului din baze de date, tabele și răspunsuri ale utilizatorilor. Cu tot acest progres general, pare ciudat faptul că mecanismele de legătură nu au progresat. V-ați gândit vreodată cât de primitiv este mecanismul de legături al limbajului HTML? Nu vreau să pară o insultă, dar legăturile HTML sunt într-adevăr tâmpite. Puteți să folosiți o gamă amețitoare de instrumente pentru a crea codul pentru paginile Web (inclusiv tehnologiile HTML, Java, JavaScript și ActiveX, pentru a numi câteva), dar, când trebuie create legăturile între pagini, sunteți adus în situația de a le adăuga manual. Nu este surprinzător faptul că legăturile întrerupte reprezintă probabil cea mai mare problemă a unui administrator Web și aproape sigur solicită cea mai mare parte a activității acestuia.

Deoarece legăturile sunt înglobate direct în documentele HTML, nu numai că este aproape imposibil să le întreții, dar și o cantitate incredibilă de date și informații este situată în afara posibilităților acestor legături. Da, puteți lega paginile HTML de fișiere grafice și puteți folosi hărțile imagine pentru a lega grafica din paginile dumneavoastră HTML de alte pagini HTML. În oricare dintre cazuri, pentru a face legătura, trebuie să aveți permisiune de

scriere, deoarece trebuie să modificați legătura. Încercați să faceți o legătură de pe un material numai cu permisiune de citire (cum este discul CD-ROM) – acest lucru nu este posibil.

Dacă doriți să faceți legătura cu un anumit punct din alt document HTML, trebuie să încărcați acel document în întregime și apoi să vă deplasați în punctul pe care l-ați adresat. Numai această cerință este responsabilă pentru irosirea considerabilă a capacității Internetului. Doriți să vă legați la alt document, dar doriți să mențineți deschis documentul curent? Ei bine, puteți deschide altă fereastră a browserului, sau puteți deschide unul sau mai multe cadre. În acest mod acceptați consecințele complicării navigării și probabil al pierderii de către cititori a orientării în sit.

Limbajul XML a schimbat toate acestea prin deschiderea unor noi și interesante posibilități pentru legarea obiectelor. Când se face legarea la un document XML (ceea ce înseamnă că ar fi posibil să se folosească aceste mecanisme de legare în documentele care nu sunt tip XML, chiar și în documente HTML, pentru a apela documente XML), se face extinderea radicală a legăturilor care pot fi create, cu următoarele caracteristici:

- XML permite utilizarea legăturilor cu capete multiple, fie de tip una-la-multe, fie de tip multe-la-multe.
- Legăturile XML sunt bidirecționale (acest lucru este foarte asemănător butonului Back din browserul Web).
- Capetele legăturilor XML (începutul sau sfârșitul) nu trebuie să fie localizate fizic în interiorul unuia dintre documentele (sau obiectele) legate. Acest lucru determină următoarele posibilități:
  - Legăturile pot fi adăugate în documente după ce acestea au fost create sau adăugate în situl Web curent, fără a vă forța să vă întoarceți și să editați manual fiecare document.
  - Legăturile pot fi generate automat (prin software) și, ca o consecință logică, ele pot fi generate și în mod dinamic.
  - Puteți crea documente „virtuale", cum sunt indexurile, care sunt formate din legături la alte documente.
- Legăturile XML se pot comporta într-un mod mult mai sofisticat. Folosind atributul SHOW, puteți determina modul de afișare al țintei unei legături. Aceasta poate fi înglobată în documentul sursă, poate înlocui documentul sursă (cum se întâmplă acum cu un document HTML simplu) sau poate să-l deschidă într-un „context" nou (un nou cadru sau o nouă fereastră al browserului). Activarea legăturii poate fi specificată fie manual (ca în cazul legăturilor HTML), fie automat, imediat ce sunt întâlnite.
- Legăturile XML pot selecta părți ale documentelor XML. Folosind o sintaxă sofisticată pentru pointerii extinși, bazată pe o aplicație SGML din domeniul academic, numită Inițiativa de codare a textului (Text Encoding Initiative – TEI), puteți utiliza structura și ierarhia elementelor pentru a alege în mod selectiv copiii elementelor sau părinții acestora, sau să numărați aparițiile și să alegeți numai anumite elemente.
- Legăturile XML pot purta o etichetă sau un nume (folosind atributele ROLE și LABEL) pentru a identifica scopul și natura lor. Folosirea acestei proprietăți nu este evidentă imediat, dar imaginați-vă că o folosiți combinată cu posibilitatea de a da semnificație semantică marcării. Puteți, de exemplu, să folosiți elementele XML la identificarea conținutului unei baze de date, iar atributele legăturii la descrierea relațiilor dintre ele. Folosind Java, sau alt limbaj de procesare, apoi puteți interoga o bază de date și salva rezultatele într-un document XML structurat. Deoarece documentul XML ar fi autodescriptiv, vă puteți rafina interogarea datelor locale și să selectați iarăși datele, fără a mai fi necesar să faceți mai multe interogări a bazei de date. În acest moment, ar

însemna că puteți repartiza sarcinile de procesare între serverul Web și aplicațiile locale, crescând în mod dramatic viteza de răspuns.

În această comparație dintre XML și HTML, am putut prezenta numai în mod superficial aspectele generale ale mecanismelor de legătură caracteristice XML. Totuși, în Capitolul 38, voi demonstra cum se pot utiliza aceste instrumente puternice, prin câteva exemple.

Stiluri inteligente

Veți putea utiliza foile de stil în cascadă (CSS1) în documentele XML în același mod în care sunt folosite în documentele HTML. De fapt, foile CSS1 vor fi probabil folosite pe scară largă pentru documentele XML simple, dar acest lucru are anumite limite când încercați să le folosiți pentru aplicații complexe. Totuși, veți putea folosi limbajul XSL (Limbajul de stil extensibil – Extensible Style Language) pentru a formata documentele HTML.

La fel cum limbajul XML introduce propria sintaxă și propriul limbaj pentru legături, a introdus și propriul limbaj pentru stil: XSL. Pe scurt, XSL are la bază Limbajul pentru semantica și specificațiile stilului documentelor (Document Style Semantics and Specification Language – DSSSL, ISO/OEC 10179:1996), care a fost dezvoltat pentru a specifica formatul documentelor structurate (SGML). În plus, acesta cuprinde forma oficială, standard a limbajului JavaScript, cunoscută sub numele de ECMA (Asociația Europeană pentru Standardizarea Informației și Sisteme de Comunicații – the European Association for Standardizing Information and Communication Systems), standardul 262 „ECMAScript". Pentru mai multe detalii, consultați Capitolul 36, „Ce este limbajul XML?" și Capitolul 39, „Să înțelegem limbajul XSL".

NOTĂ

XSL este încă într-o fază inițială de dezvoltare; în momentul apariției acestei cărți, standardul nu a depășit stadiul unui document pentru discuții. Se aude că Microsoft pregătește o propunere pentru limbajul de stil al XML și nu există nici o garanție că oricare dintre părțile actuale ale propunerii vor rămâne neschimbate.

XSL este tot atât de diferit de CSS1 precum este XML față de HTML. Esențial este faptul că XSL este un limbaj complet de programare, care vă permite să realizați procese cu context condițional, complexe și dependente de structură, într-un document XML. De exemplu, puteți face următoarele lucruri folosind XSL :

- Să extrageți elemente dintr-un anumit loc al unui document și să le folosiți în alt loc. Această facilitate permite, de exemplu, să se genereze în mod automat tabele de conținut, indexuri și referințe încrucișate.
- Să se formateze în funcție de relația structurală a elementelor. Acest lucru poate fi făcut în CSS1 prin folosirea pseudoelementelor, dar în XSL puteți formata, de exemplu, paragrafe alternante în stiluri diferite.
- Să calculați și să păstrați variabilele în memorie. Această facilitate vă permite să transmiteți valori ale parametrilor când procesați foi de stil și să generați diferite elemente, precum secvențe de autonumărare, numerele paginilor, anteturi și subsoluri pentru pagini.

În afară de faptul că este „doar" un limbaj de programare (unii ar putea argumenta că Java sau JavaScript, sau chiar XML ar putea îndeplini aceste sarcini tot la fel de ușor), XSL este complet adaptat nevoilor de editare profesională. Pe de altă parte, CSS1 este mult mai adaptat pentru afișarea pe ecran a codului HTML decât pe foaia de hârtie. XSL cuprinde facilități pentru următoarele funcții:

- Tipărirea pe mai multe direcții (de la stânga la dreapta, de la dreapta la stânga, de sus în jos); direcțiile pot fi combinate în mod arbitrar
- Tipărirea mai multor coloane, inclusiv intervale de coloane și machetarea sincronizată pe coloane
- Obiecte text flotante (note, note de subsol ș.a.m.d.)
- Zone cu text rotit
- Elemente atașate pe margini (texte suplimentare, marcaje de revizuire ș.a.m.d.)

Este dificil să se exagereze puterea limbajului XSL. Pentru a-i acorda atenția cuvenită ar fi necesară o carte separată. Totuși, rămâne deschisă problema dacă forma finală a limbajului de stil al XML va arăta asemănător cu limbajul XSL prezentat aici. Pentru moment, o mare parte din limbajul XSL a fost deja implementată în pachete de software (ca de exemplu Jade) ca o formă prescurtată a limbajului părinte, DSSSL, și poate fi folosită acum pentru conversia documentelor HTML în alte formate. În Capitolul 39 sunt descrise unele exemple ale acestor conversii.

Nu poate Java, JavaScript sau ActiveX să rezolve problema?

Folosind programarea în Java, Java Script, ActiveX, CGI sau o combinație de mecanisme de codare sau pentru script, puteți probabil suplini limitările HTML pe care XML este destinat să le rezolve sau puteți realiza multe din facilitățile pe care le introduce XML. Folosirea scripturilor înglobate a fost plasată pe un teren mai sigur prin adăugarea elementului OBJECT în HTML 4.0, iar scripturile înglobate oferă un mijloc puternic de extindere a HTML dincolo de limitele pe care le-au visat dezvoltatorii săi. Problema nu este dacă acest lucru poate fi făcut prin înglobarea codului în documente; problema este dacă *trebuie* făcut acest lucru.

Noțiunea centrală a limbajului SGML este separarea conținutului de prezentare. Modul de prezentare a informației a fost întotdeauna un impediment major pentru portabilitatea textului. Fie că cealaltă platformă nu are fonturile, ori are un mecanism de codare diferit, ori software-ul nu recunoaște codurile folosite.

Cu ani în urmă, oamenii acceptau mai ușor faptul că un document Macintosh nu poate fi văzut pe PC sau că un document editat cu produsul DisplayWrite al IBM nu poate fi citit cu programul Manuscript al firmei Lotus. Lucrurile au evoluat spre mai rău în ultimul timp, când nu există nici măcar garanția că un document poate fi citit cu diferitele versiuni ale aceluiași pachet de software. A fost aproape imposibil să se impună tuturor producătorilor de software să se pună de acord în ceea ce privește folosirea unui singur format sau a unui singur format interschimbabil. Trebuia elaborat un format neutru, așa că IBM (care este, chiar și acum, cel mai mare editor de informație tipărită) a venit cu GML, care s-a transformat în SGML.

Premisa de bază a limbajului SGML este că aspectul unei porțiuni de text (de exemplu, dacă este editat cu caractere aldine, cu fonturi mai mari sau pe o linie separată) este de obicei funcție de scopul acestuia în cadrul documentului. Funcția acestuia poate fi exprimată potrivit relației cu celelalte părți ale documentului. Dacă puteți identifica ceva ca fiind antet, puteți lăsa platforma locală să rezolve problema găsirii caracteristicilor de afișare potrivite.

Ducând această teorie mai departe, prin identificarea structurii documentului puteți permite cititorilor să aleagă caracteristicile de prezentare cele mai potrivite cerințelor lor (Braille, audio, caractere foarte mari ș.a.m.d.).

Înglobarea codului în documentele HTML pentru a obține „efecte de prezentare" este echivalentă cu întoarcerea ceasului înapoi, renunțarea la tot ceea SGML s-a luptat să obțină și, prin urmare, subminarea în mare parte a ceea ce HTML a fost dezvoltat să facă. În ultimul rând, transformarea documentelor în fragmente de cod ar putea fi un lucru bun din punctul de vedere al unui programator, dar pentru restul dintre noi, care nu suntem programatori, transferă misiunea de realizare a documentelor HTML înapoi la îndemâna celor cu pregătire specială. Mai rău, acest lucru amenință serios portabilitatea documentelor HTML și le face foarte dificile în ceea ce privește întreținerea.

Rezumat

În acest capitol am încercat să vă prezint o imagine sumară a diferențelor principale dintre HTML și XML. Deoarece HTML este o aplicație a SGML, iar XML o versiune a SGML, comparația este puțin „necinstită", comparându-se o aplicație cu o trusă de instrumente a unei aplicații. Totuși, prin compararea unor caracteristici principale ale HTML și ale unei aplicații XML cu caracter general, ar trebui să puteți aprecia într-o oarecare măsură avantajele nete pe care le prezintă XML.

M-am concentrat asupra caracteristicilor în termeni generali și am trecut peste detalii (cum ar fi cerințele XML ca toate etichetele elementelor să fie imbricate corespunzător și toate valorile atributelor să fie între ghilimele), cu intenția de a oferi o mai bună imagine globală a diferențelor dintre XML și HTML. Pentru aspectele detaliate, reveniți la Capitolul 36.

Am trecut de asemenea în mod intenționat peste anumite caracteristici ale limbajului XML, cum ar fi compatibilitatea cu aplicațiile internaționale prin folosirea Unicode (ISO 10646), deoarece aceste caracteristici au fost deja sau vor fi în curând atinse de perfecționări ale HTML. În Capitolele 38 și 39, veți avea șansa de a vedea exemple practice în care se folosesc aceste caracteristici.

Capitolul 38
Utilizarea limbajului XML

de Simon North

În acest capitol:

- **Formatat corespunzător sau valid? 590**
- **Crearea unei definiții DTD 591**
- **Interiorul documentului XML 595**
- **Legăturile XML 598**

Ați întâlnit partea tehnică a limbajului XML în Capitolul 36, „Ce este limbajul XML?" și 37, „Comparație între limbajele XML și HTML" (și o veți vedea din nou în Anexa D, „Scurtă prezentare a limbajului XML"). În acest capitol vă voi duce departe de discuțiile privind sintaxa și semantica, într-o călătorie unde veți vedea utilizările practice ale limbajului XML.

Cu ajutorul unor exemple bogate, veți studia următoarele subiecte:

- Când este necesară sau nu definiția DTD
- Principiile de bază ale proiectării definiției DTD în XML
- Modelarea elementelor și atributelor
- Utilizarea entităților text ca prescurtări
- Utilizarea entităților parametru ca prescurtări și pentru secțiunile condiționale
- Utilizarea entităților binare externe ca declarații de notație
- Utilizarea și riscurile utilizării unor caracteristici ale legăturilor XML

Formatat corespunzător sau valid?

În Capitolul 36, am prezentat definițiile DTD (document type definitions) și ce conțin acestea, dar după toate informațiile prezentate se poate pune întrebarea: avem nevoie de definiția DTD? este o întrebare care se poate pune foarte ușor, dar este foarte greu de răspuns la ea. Ușor sau greu, trebuie răspuns la întrebare înainte de a merge mai departe, deoarece acest răspuns determină în mare măsură activitatea pe care trebuie să o desfășurați mai departe.

Răspunsul ușor, pentru a cita specificația XML, este :„numai dacă subsetul extern conține un atribut neprecizat cu valori prestabilite, referințe ale entității sau dacă spațiul alb apare în mod direct în cadrul tipurilor de elemente care au conținut de elemente". Or, mai clar spus, numai dacă partea externă a definiției DTD conține atribute care nu sunt definite în document, dar au valori prestabilite, dacă conțin referințe ale entității sau dacă ați introdus spații în structura documentului în locurile unde s-ar afla de obicei elementele.

Răspunsul dificil este, bineînțeles, mult mai greu de dat; acesta se reduce mai degrabă la problema cântăririi unor considerații decât la aplicarea unor reguli prefabricate.

Întrebați-vă, de exemplu, ce tip de verificare a documentelor doriți să faceți. Dacă documentele trebuie să fie generate de un program, există șanse ca dumneavoastră să puteți adapta programul astfel încât să genereze corect codul XML, astfel încât validitatea să nu mai fie o problemă.

De asemenea, gândiți-vă ce veți face cu documentele după ce le-ați creat. Dacă veți refolosi conținutul sau îl veți procesa într-un mod oarecare (cum ar fi conversia în HTML), definiția DTD vă oferă o structură formală care poate fi procesată extern pentru a se obține reguli de procesare a documentelor.

Nu vă lăsați descurajat de complexitatea aparentă a definiției DTD; definiția DTD trebuie să fie atât de complexă cât o doriți dumneavoastră sau cât este nevoie să fie. Pe de altă parte, poate să fie atât de simplă cât o doriți dumneavoastră sau cât este nevoie să fie, cuprinzând poate doar câteva elemente. De exemplu, este în circulație o definiție DTD propusă pentru documentele cu întrebări frecvente din Internet (Internet FAQ) și aceasta are numai trei sau patru elemente; un element `Q` pentru întrebări, un element `A` pentru răspunsuri și încă alte câteva pentru a realiza legarea la alte întrebări și răspunsuri.

Pe măsură ce limbajul XML devine din ce în ce mai popular, ar trebui să fie disponibile pentru utilizare definiții DTD pentru XML, așa cum sunt și pentru SGML

Crearea unei definiții DTD

O dată cântărite avantajele și dezavantajele existenței definiției DTD, să presupunem că v-ați decis să scrieți o definiție DTD pentru documentul dumneavoastră. În următoarele subcapitole, voi sublinia câteva din sarcinile și problemele pe care trebuie să le aveți în vedere când creați o definiție DTD. Totuși, înainte de a vă grăbi să începeți scrierea codului, căutați bine în Web, pentru a vedea dacă nu cumva puteți găsi altă definiție DTD, care ar putea folosi scopului dumneavoastră. O definiție DTD existentă, cu puține modificări, vă poate ajuta să economisiți timpul necesar realizării unor lucruri deja realizate de alții – și poate realizate mai bine. Sau puteți să folosiți definițiile DTD ale altor persoane doar ca un material de pregătire; încercați să urmați logica concepției și să înțelegeți deciziile de proiectare care au jucat un rol în acest context.

Definiția DTD este un fișier ASCII simplu – puteți folosi orice editor sau procesor de texte doriți pentru a-l crea, cu condiția să nu uitați să-l salvați în format ASCII. Nu există încă nici un pachet de dezvoltare pentru DTD în limbajul XML. Există totuși unele pachete specializate în dezvoltarea DTD pentru limbajul SGML. Acestea se întind de la Near & Far, un performant pachet pentru modelare grafică a DTD al firmei Microstar, până la pachetele shareware gratuite, cum sunt de exemplu EasyDTD, care convertesc fișierul ASCII ales într-o structură de element DTD corect marcat.

Alegerea elementelor (maparea arborelui)

Dacă proiectați o definiție DTD pentru un tip de document care există deja în altă formă, prima sarcină a dumneavoastră este să analizați documentul pentru a încerca să identificați elementele din tipul documentului. Aveți grijă să analizați o colecție reprezentativă de documente de tipul cerut! Multe proiecte s-au împotmolit când s-a încercat adaptarea lor la cazuri speciale, încercarea de generalizare a unei convenții obișnuite pentru toate variantele unui anumit tip de document dovedindu-se imposibilă. S-ar putea să descoperiți că varietatea conținutului și structurii tipului de document v-ar putea forța să concepeți mai multe definiții DTD.

Principala dumneavoastră grijă este să încercați să sintetizați un model general al obiectelor informații din documente. Gândiți-vă la fiecare document ca la o bază de date și modul dumneavoastră de gândire va evolua în direcția bună, deoarece proiectarea corectă a DTD are multe părți comune cu cea a bazelor de date. Este important să nu uitați că dacă nu ați identificat unele informații din DTD, fie ele elemente sau atribute, nu le veți putea procesa prin interogare, formatare ș.a.m.d.

Dacă reușiți să identificați mai degrabă funcția, decât aspectul unei porțiuni de text, înseamnă că vă păstrați aproape de filozofia SGML – aceea de a separa conținutul de prezentare. De exemplu, în HTML o listă (OL sau UL) este compusă din elementele listei (LI). Această definiție leagă strâns un element LI de contextul acestuia. În definiția dumneavoastră DTD din limbajul XML puteți aborda problema într-un mod mult mai abstract și să îmbunătățiți „modelul conținutului". Să presupunem că elementul de bază în care trebuie să apară textul respectiv (datele caracter) este elementul PARA. UL ar fi definit în mod normal ca având în conținutul său unul sau mai multe elemente ale listei, după cum se vede mai jos:

```
<!ELEMENT UL (LI)+ >
<!ELEMENT LI (#PCDATA) >
```

Folosind elementul PARA, puteți totuși subîmpărți elementele listei:

```
<!ELEMENT UL (LI)+ >
<!ELEMENT LI (PARA)+ >
<!ELEMENT PARA (#PCDATA) >
```

Avantajul acestei abordări este că elementul PARA, ca bloc de text, poate fi acum mutat cu ușurință în și dintr-un element listă, iar dumneavoastră puteți foarte ușor să compuneți sau să divizați elementele listei. Rezultatul final face ca elementele PARA să fie mult mai ușor de manipulat și refolosit în acest document sau în alte documente.

Fiind alese elementele pe care doriți să le identificați, plasați-le într-o structură tip arbore. O cerință imperativă a limbajului XML este următoarea: trebuie să fie un element care să servească drept rădăcină sau părinte al întregului arbore, iar numele elementului trebuie să fie același cu numele DTD.

Veți constata probabil, în special la început, că trasarea unui arbore al elementelor este de mare ajutor pentru construcția DTD. Apoi, după ce abandonați aceste diagrame ca nemaifiind necesare, vă veți reîntoarce la desenarea lor ca la cea mai bună metodă de explicare a definiției DTD altor persoane.

Evitarea ambiguității

Când construiți un arbore, asigurați-vă că acesta nu este ambiguu. Procesoarele XML sunt destinate să fie ușor de creat și rapide, ceea ce înseamnă și faptul că nu sunt foarte sofisticate. În mod particular, ele nu sunt destinate (și probabil nici nu pot) să se uite înainte și să ghicească elementul următor și dacă ceea ce într-adevăr vede este permis. A fi lipsit de ambiguitate înseamnă deci a fi clar și explicit. Un procesor XML simplu ar trebui să fie capabil să proceseze DTD și să confirme rapid și ușor dacă un document XML scris pentru acel DTD este valid.

Modelele cu conținut ambiguu pot fi o problemă majoră pentru limbajul SGML, în cea mai mare parte deoarece SGML vă permite să definiți elemente într-o DTD astfel încât este posibil să omiteți etichetele de început și sfârșit. Analizorul SGML cunoaște din context unde ar fi trebuit să apară o etichetă de început sau sfârșit și poate deduce din context ce element este procesat în acel moment. În XML, lucrurile sunt ceva mai simple datorită faptului că această metodă simplificată, numită *optimizarea etichetei*, nu este permisă; fiecare element XML trebuie să aibă o etichetă de început și una de sfârșit. Acest lucru nu înseamnă totuși că problema ambiguității a dispărut.

În XML, trebuie să fiți atent la elementele cu „conținut amestecat", care pot conține date caracter și elemente, ca în exemplul de mai jos:

```
<!ELEMENT P (#PCDATA | A | EM | STRONG | TT)* >
```


Observați că, după cum este cerut de sintaxa XML, datele caracter (PCDATA) sunt totdeauna pe prima poziție, iar modelul conținutului folosește numai condiția OR (:). Acest lucru ar putea părea fără importanță, dar gândiți-vă ce se întâmplă când procesorul XML vede următoarele:

```
<P>
     <EM>Aici începe textul</EM>  <A>si aici
este o ancora</A></P>
```

Procesorul XML va vedea caracterul saltul la rând nou (sau saltul la rând nou și revenire la începutul rândului), înainte de elementul EM și va vedea spațiul alb dintre elementele EM și A! Dacă nu păstrați regulile, veți avea multe neplăceri.

NOTĂ

Problema spațiului alb, un subiect sensibil în cercurile XML, nu pare să aibă o soluție definitivă. În HTML, spațiul alb este pur și simplu ignorat; în XML, el poate fi ignorat sau luat în seamă. Cea mai bună metodă de lucru este subiectul multor discuții aprinse.

Elemente sau atribute?

Altă întrebare – una care a produs multe bătăi de cap în cercurile SGML și XML și pe care va trebui să o aveți și dumneavoastră în vedere – este legată de folosirea elementelor sau atributelor.

Următorul exemplu este o definiție DTD completă, dar foarte simplă, pentru mesaje de poștă electronică:

```
<!element email (to, from, date, subject, body) >
<!element to       (#PCDATA)>
<!element from     (#PCDATA)>
<!element date     (#PCDATA)>
<!element subject  (#PCDATA)>
<!element body     (para)+>
<!element para     (#PCDATA)>
```

DTD identifică pur și simplu obiectele de bază ale informației pe care vă așteptați să le găsiți într-un mesaj de poștă electronică. Dar face acest lucru? Datele calendaristice pot fi scrise în moduri diferite. În Marea Britanie, se obișnuiește ca ziua să se scrie înaintea lunii; în America, se scrie mai întâi luna de obicei.

Cea mai importantă problemă cu care vă confruntați este să stabiliți cum va fi folosită informația privind data. Pentru afișarea mesajului de poștă electronică, probabil doriți ca data să fie într-o formă prietenoasă pentru utilizator, cum este aceasta:

```
August 12th, 1997
```

Totuși, din puct de vedere al modelării informației, doriți probabil să puteți găsi toate mesajele trimise înainte sau după o anumită dată sau într-un interval de timp. Pentru acest lucru, probabil preferați să exprimați data într-o formă mai adaptată procesării de către calculator, ca în acest exemplu:

```
19970812
```

La prima vedere, cele două cerințe par a fi în conflict, dar există multe alte căi de a le împăca pe amândouă. Cea mai simplă cale de a satisface ambele cerințe este să *descompunem* data în subelemente care conțin componente ale datei și să transformăm elementele datei într-un element „container":

```
<!ELEMENT date (year, month, day) >
<!ELEMENT year (#PCDATA)>
<!ELEMENT month (#PCDATA)>
<!ELEMENT day (#PCDATA)>
```

În acest caz, data va fi marcată după cum urmează:

```
<date><year>1997</year><month>8</month><day>12</day></date>
```

Pentru afișare, veți cere foii de stil să reordoneze componentele datei (mai întâi `month`, apoi `day` și apoi `year`), să convertească numărul lunii în cuvânt (`August`) și să adauge un sufix potrivit (`st`, `nd`, `rd`, `th`) și o virgulă la `day`:

```
August 12th, 1997
```

O strategie alternativă este să se folosească *atribute* pentru a include una dintre formele în care este solicitată informația. În acest caz, este probabil mai ușor să avem forma prietenoasă a datei calendaristice în conținutul propriu-zis de date și să dăm forma numerică sub forma valorii atributului `NUMBERFORM`.

Pentru a face acest lucru, trebuie să declarați tipul elementului și o listă de atribute pentru `date`:

```
<!ELEMENT DATE (#PCDATA)>
<!ATTLIST date
          NUMBERFORM CDATA #IMPLIED>
```

Data va fi introdusă după cum urmează:

```
<DATE NUMBERFORM="19970812">August 12th 1997</DATE>
```

Observați că valoarea prestabilită pentru `NUMBERFORM` este `#IMPLIED`. Dacă în loc de aceasta ați specificat `# REQUIRED` utilizatorii vor fi nevoiți să introducă o valoare pentru atribut sub formă numerică (ceea ce s-ar putea să fie o idee bună).

Care metodă ar trebui aleasă? În ultimul rând, dumneavoastră sunteți singurul care într-adevăr puteți decide acest lucru, deoarece sunteți suficient de familiarizat cu propria aplicație pentru a lua o decizie în cunoștință de cauză. Consensul general între proiectanții XML este că, dacă nu există restricții evidente ale aplicației, alegerea este în principal o problemă legată de preferințele personale.

Opinia mea este că e puțin mai ușor să validezi și să întărești introducerea și conținutul atributelor, decât conținutul elementelor. XML (și SGML) nu au nici o cale de asigurare a faptului că un element are conținut sau nu este lipsit de sens, dar puteți avea cel puțin un oarecare control asupra prezenței și conținutului valorii unui atribut. Pe de altă parte, ca în acest exemplu, ar trebui să acceptați riscul ca valoarea atributului și valoarea exprimată în conținutul elementului să nu fie compatibile între ele. Mai departe, evident, utilizatorul/scriitorul ar trebui să introducă aceeași informație de două ori.

Entitățile în definiția DTD

Entitățile reprezintă probabil unul din cele mai subevaluate, dar folositoare, elemente din SGML. În XML, ele sunt mai la îndemână. Există trei tipuri principale de entități:

- **Entități text**: Identificate prin &, aceste entități pot fi folosite atât în definiții DTD, cât și în documente XML.
- **Entități parametru**: Identificate prin %, pot fi folosite numai în definițiile DTD din XML.
- **Entități binare**: Aceste entități pot fi folosite numai în documente XML.

Mă voi reîntoarce la problema entităților text și binare mai târziu, în cadrul discuțiilor privind documentele XML (vezi „Interiorul documentului XML"). Acum să aruncăm o privire asupra entităților parametru și cât de utile pot fi ele într-o definiție DTD din XML.

Entitățile parametru pot fi folosite ca un fel de macrolimbaj simplificator pentru marcare. Dacă veți folosi aceleași segment de marcare de mai multe ori în definiția DTD, îl puteți declara ca fiind entitate parametru:


```
<!entity % inline
  * (#pcdata | f| x| %emph; |sq| %xref | %index )* ">
```

Apoi, ori de câte ori doriți să repetați segmentul de marcare, faceți simplu referire la el, după cum se vede:

```
<!element footnote (%inline;)>
<!element subject (%inline;) >
<!element closing (%inline;) >
<!element cc (%inline;) +(newline) >
<!element encl (%inline;) +(newline) >
```

Entitățile parametru pot fi folosite și pentru marcarea provizorie care e formată din secțiuni „condiționale". Să presupunem că dezvoltați un document XML pentru un grup de scriitori. O dată ce documentul a fost terminat, doriți să-l validați complet. Totuși pe timpul cât lucrați la el, doriți ca structura să fie puțin mai flexibilă. Pentru proiectul preliminar este posibil să aveți următoarea declarație:

```
<!ELEMENT document (title, abstract?, intro?, section*)>
```

În limbaj simplu, această declarație spune: trebuie să fie un titlu, urmat de un rezumat opțional, o introducere opțională și un număr de capitole (inclusiv niciunul). Acest lucru este ideal pentru proiecte – dispuneți de atâta loc cât aveți nevoie.

Totuși, pentru varianta finală a documentului, trebuie să fiți riguros:

```
<!ELEMENT document (title, abstract, intro, section+)>
```

Această declarație este interpretată astfel: trebuie să fie un titlu, urmat de un rezumat, o introducere și cel puțin un capitol. Acum ați abordat ambele situații – dar mai este ceva. Puteți să puneți ambele declarații în definiția DTD și să folosiți o entitate parametru ca pe un „comutator" care să pornească versiunea dorită și să oprească versiunea nedorită:

```
<!ENTITY % draft "INCLUDE">
<!ENTITY % final "IGNORE">

<![ %draft; [
<!ELEMENT document (title, abstract?, intro?, section*)>
]]>

<![ %final; [
<!ELEMENT document (title, abstract, intro, section+)>
]]>
```

Tot ce aveți de făcut pentru a activa versiunea pe care o doriți este să introduceți în declarația pentru versiunea dorită valoarea `INCLUDE`, iar în cealaltă valoarea `IGNORE`.

Interiorul documentului XML

Până acum, în acest capitol ați văzut numai definiția DTD din limbajul XML, dar aceasta este numai o parte a problemei. Sunt multe de făcut chiar în cadrul documentului XML.

După cum probabil vă amintiți din Capitolul 36, definiția DTD poate fi divizată în două subseturi: subsetul extern (păstrat într-un document separat) și subsetul intern, care se află la începutul respectivului document XML:

```
<!DOCTYPE EMAIL SYSTEM "EMAIL.DTD" [
<!-- subsetul intern DTD apare aici -->
]>
<!--iar continutul documentului incepe aici -->
```

În subcapitolele următoare, veți vedea ce se poate face cu entitățile în subsetul intern DTD.

Entitățile text

Entitățile XML tip text se împart în două clase: entități interne, în cazul cărora conținutul entităților apare în declarație și entități externe, în cazul cărora conținutul apare în altă parte (de exemplu într-un fișier extern)

Entitățile text interne

Entitățile text interne sunt puțin asemănătoare cu entitățile parametru pe care le-am prezentat mai devreme (dar observați sintaxa diferită) și pot fi utilizate ca un fel de prescurtare:

```
<!ENTITY W3C.home
'<a href="http://www.w3c.org/">The W3C Consortium</a>'>
<!ENTITY W3C.XML
'<a href="http://www.w3c.org/Activity/XML/">The W3C XML Page</a>'>
...
```

Cu această declarație, puteți folosi aceste entități ca o prescurtare pentru legăturile A din textul documentului dumneavoastră XML:

```
<P>For an overview of the W3C's work see &W3C.home;. For
details of their work on XML see &W3C.XML;. ...
```

Acest cod va fi procesat ca și cum ați fi introdus în întregime elementul A de fiecare dată:

```
<P>For an overview of the W3C's work see
<a href="http://www.w3c.org/"
>The W3C Consortium</a>. For details of
their work on XML see
<a href="http://www.w3c.org/Activity/XML/">The
W3C XML Page</a>. ...
```

Entități text externe

În loc să declarați entitățile în interiorul documentului XML, le puteți pune într-un fișier separat, dar *nu* în DTD. O astfel de separare a entităților vă oferă o metodă convenabilă de păstrare a tuturor legăturilor de la oricâte documente XML într-un singur loc. Bineînțeles că această metodă nu va opri alte situri Web să-și schimbe adresele URL, dar va ușura durearea de cap dată de verificarea faptului că toate legăturile la acestea sunt actualizate.

Creați un fișier care conține doar declarațiile entității și dați-i numele `/links/ext_lnks.ent`. Atunci conținutul fișierului ar arăta exact ca în declarația originală pe care am descris-o anterior:

```
<!ENTITY W3C.home
'<a href="http://www.w3c.org/">The W3C Consortium</a>'>
<!ENTITY W3C.XML
'<a href="http://www.w3c.org/Activity/XML/">The W3C XML Page</a>'>
...
```

Apoi, declarați acest fișier în locul entităților din subsetul intern DTD din XML. Observați că deși am folosit un identificator SYSTEM urmat de un nume de fișier în exemplul precedent, aș fi putut la fel de ușor să fi folosit o adresă URL. Nu uitați că trebuie să faceți și referința la entitate (dacă uitați referința entității, conținutul acesteia va fi omis):

```
<!DOCTYPE EMAIL SYSTEM "EMAIL.DTD" [

<!--declarati fisierul ca entitate -->
<!ENTITY % external.links SYSTEM "http://myhome.com/ext_lnks.ent">

<!--si acum faceti referinta -->
```



```
%external.links;

<!--urmeaza alt subset local DTD -->
]>

<!--iar acum urmeaza documentul XML propriu-zis -->
```

Acum puteți folosi aceste entități ca și cum le-ați fi declarat local în subsetul intern DTD:

```
<P>For an overview of the W3C's work see &W3C.home;. For
details of their work on XML see &W3C.XML;. ...
```

Avantajul acestei metode este că toate legăturile externe pot fi ținute într-un singur loc, iar dumneavoastră puteți să declarați fiecare legătură numai o dată. Dacă o adresă URL se schimbă – așa cum se întâmplă adesea – trebuie actualizată numai o dată. Atunci puteți fi sigur că toate paginile dumneavoastră cu legătură la acea adresă URL vor continua să funcționeze.

Entități binare externe

Entitățile binare externe par destul de exotice; de fapt, ele sunt numai imagini binare pe care le-ați folosit probabil pentru referința elementului IMG din HTML.

XML e ceva mai pretențios în sintaxă la declararea entităților binare externe; trebuie să urmați aceste două reguli:

- Trebuie să declarați o notație.
- Acestea pot fi referite numai prin valorile atributelor.

Există două moduri de a identifica o notație:

- Prin identificarea după nume a formatului entității binare externe:

```
<!DOCTYPE HTML SYSTEM "HTML4.DTD" [
<!NOTATION GIF SYSTEM "Compuserve/Unisys Graphics Format">
```

- Prin identificarea după numele aplicației căreia i se adresează instrucțiunile de procesare:

```
<!DOCTYPE HTML SYSTEM "HTML4.DTD" [
<!NOTATION GIF SYSTEM "/gsworks.exe">
```

Indiferent de metoda aleasă, o dată ce ați declarat notația, o folosiți în exact același fel când declarați entitatea binară externă:

```
<!DOCTYPE HTML SYSTEM "HTML4.DTD" [

<!--aici se declara notatia -->
<!NOTATION GIF SYSTEM "/gsworks.exe">

<!--si aici declaram entitatea -->
<!ENTITY lefthand SYSTEM "hand_pointing_left.gif" NDATA GIF>
]>
```

Acum, spre deosebire de HTML, faceți referința la grafic (cel puțin în acest exemplu) prin specificarea atributului SRC care să indice către entitate:

```
<PARA>This is the image we should see <FIG SRC="lefthand"/>
```

S-ar părea că este necesar un efort considerabil ca să se definească o legătură la un grafic și poate că așa și este. Gândiți-vă totuși la flexibilitatea pe care o poate introduce această metodă. În loc să vă îngrijorați dacă browserul a fost setat să recunoască tipul corect MIME (care vă conectează la o anumită convenție de denumire) și să nădăjduiți că au fost urmate convențiile corecte de nume, formatul graficului la care faceți referința este acum ceea ce spuneți că este, indiferent de modul cum a fost denumit fișierul.

După cum ați văzut, deși limbajul XML nu vă solicită să aveți DTD, forța deplină care deosebește XML de HTML stă în caracteristicile puternice pe care le puteți îngloba în DTD.

Definițiile DTD pot constitui un subiect extrem de complex și, în cercurile SGML, dezvoltarea acestora este aproape o disciplină de sine stătătoare. În limbajul XML, așa cum s-ar putea să constatați după ce vă veți familiariza cu el, a fost găsit un admirabil compromis între putere și ușurința dezvoltării.

Legăturile XML

Pentru a încheia acest capitol asupra folosirii XML, aș dori să vă ofer o scurtă privire asupra unui lucru care într-adevăr face ca limbajul XML să fie special: legăturile. Legăturile XML sunt cele care-l deosebesc într-adevăr atât de HTML, cât și de SGML. Acestea au terminații multiple, pot avea etichete ce pot fi citite de către utilizator și au mecanisme extrem de sofisticate. Decât să alunecăm într-o discuție teoretică asupra sintaxei și mecanismelor de rezoluție (aceste lucruri le puteți găsi în Capitolul 36, Anexa D și în documentele de referință oficiale), în următoarele subcapitole voi examina doar câteva exemple practice.

Folosirea legăturilor extinse

```
<!DOCTYPE CHESSGAME [

<!-- prima data declaram elementele -->
<!ELEMENT X ANY>
<!ELEMENT L ANY>

<!-- apoi declarăm atributele lor -->
<!-- un atribut ca legatura extinsa -->
  <!ATTLIST X XML-LINK #FIXED "EXTENDED"
<!-- si unul ca adresa-->
  <!ATTLIST L XML-LINK #FIXED "LOCATOR"
]>

<PARA>Karpov began a conventional game, starting with

<X>

<!-- aceasta legatura deschide o animatie intr-o fereastra noua -->
<!-- Observati cum putem folosi rolul pentru a identifica legatura -->
<L ROLE="demonstration" LABEL="Animation"
  SHOW="NEW" HREF="/cgi-bin/qview?kpopen" />

<!-- aceasta legatura inglobeaza o imagine -->
<L ROLE="picture" LABEL="Illustration"
    SHOW="EMBED"
    HREF="pix.xml#DESCENDANT(*,FIG,CAPTION,KP OPENING)"/>

<!-- aceasta legatura selecteaza o gama din elementul -->
<!-- al carui ID este Karpov.4.1 si urmatoarele doua elemente P -->
<!-- din acelasi document -->
<L ROLE="Game Report" LABEL="Game Report"
   HREF="report.xml#ID(Karpov.4.1)..DITTO,NEXT(2,P)" />

<!-- Aceasta legatura va pozitioneaza in punctul specificat -->
<!-- din documentul HTML -->
<L ROLE="ToMove" LABEL="Move to Game transcript"
   SHOW="REPLACE" HREF="game4.html#Move1" />
a King's Pawn opening.</PARA>
```

Nici acest exemplu, nici chiar o sută de exemple nu vă pot da adevărata idee despre ceea ce într-adevăr se poate face cu puterea pe care aceste legături o pun în mâinile dumneavoastră. Explorarea acestui potențial depinde de dumneavoastră. Un sfat totuși: fiți atent – s-ar putea să nu obțineți ceea ce ați crezut că veți obține. Voi exemplifica acest lucru:


```
<PARA ID="a1">This is a simple paragraph taken from <CITE>A Famous Book</CITE>,
 showing just how clever the writer is
<EDITOR COMMENT>We beg to differ</EDITOR COMMENT>
in having read <CITE>Great Writer</CITE>.</PARA>
```

Acesta este un paragraf simplu care înglobează trei elemente, lucru obișnuit chiar și într-un document HTML. Fiți atent totuși; nu este așa de simplu pe cât pare, în special în XML!

BIBL. CENTR. UNIV. „M. EMINESCU" IAȘI INFORMATICĂ

Să vedem acum structura elementului:

```
START PARA
  CDATA
    CITE
  CDATA
    EDITOR COMMENT
  CDATA
    CITE
  CDATA
END PARA
```

Presupunem așadar că aveți o legătură la:

```
ID(a1)CHILD(4,*CDATA)
```

Ce voi obține? Punctul de la sfârșitul propoziției (al patrulea copil care e dată caracter)!

Dacă doriți al patrulea element copil, după cum e prezentat aici, veți obține elementul `EDITOR COMMENT`:

```
ID(a1)CHILD(4)
```

După cum veți descoperi cu siguranță, putere mai mare înseamnă și ocazii mai multe de a face greșeli!

Sunt multe alte căi prin care se poate arăta că legăturile XML sunt mult mai puternice decât cele HTML sau HTML (cum ar fi grupul de legături extinse). Din păcate, discuția asupra caracteristicilor acestora va trebui să fie lăsată altui autor, într-o carte care tratează subiectul mai detaliat.

Rezumat

În acest capitol ați învățat despre:

- De ce trebuie să țineți cont când stabiliți dacă trebuie să concepeți o definiție DTD și situațiile când aceasta s-ar putea să nu fie necesară
- Principiile de bază ale analizei documentelor – identificarea elementelor, a atributelor și a entităților care vor forma conținutul DTD din limbajul XML
- Cum să îndepliniți sarcina de a crea un conținut XML mai simplu prin folosirea entităților text ca un fel de prescurtare în documente
- Cum se folosesc entitățile parametru în definițiile DTD, ca o prescurtare pentru porțiunile mari de cod sau pentru construirea secțiunilor de cod condiționale pe care le puteți deschide sau închide, după cum cer circumstanțele
- Cum să se folosească entitățile binare externe și declarațiile privind notațiile pentru a încorpora elemente precum fișierele grafice și cum să se facă acest lucru într-un mod care să vă scutească de o cantitate apreciabilă de editare a legăturilor
- Cum să se folosească unele dintre caracteristicile mecanismelor de legătură ale XML – și unele dintre pericolele care vă pândesc dacă faceți acest lucru.

Capitolul 39
Să înțelegem limbajul XSL

de Simon North

În acest capitol:

- **XSL și CSS 600**
- **XSL și DSSSL 600**
- **Regulile de construcție 605**
- **Referința la o foaie de stil din limbajul XML 616**
- **Limbajul de scriptare al XSL 617**
- **Foi de stil cu structură fixă 620**

Limbajul de stil extensibil (XSL) este un limbaj pentru foaia de stil, destinat comunității Web. Este corespondentul XML pentru foile de stil în cascadă (CSS) din limbajul HTML.

În acest capitol, veți începe prin a vedea o comparație a XSL cu CSS și părintele XSL, limbajul DSSSL. Veți vedea apoi exemple practice de folosire a declarațiilor XSL în foile de stil și veți încheia prin a descoperi ce este limbajul script al XML (ECMA Script).

XSL și CSS

XSL oferă funcționalități superioare posibilităților CSS. El poate realiza funcții de editare avansate, cum ar fi reordonarea elementelor sau copierea lor în alte locații, dându-vă posibilitatea să realizați sarcini precum sunt următoarele:

- Elemente cu format selectiv bazate pe strămoșii, urmașii, poziția și unicitatea acestora
- Crearea de construcții de formatare care cuprind texte și grafice generate
- Definirea macroinstrucțiunilor de formatare reutilizabile
- Crearea foilor de stil care sunt independente de direcția de scriere
- Extinderea setului de obiecte de formatare

Foaia CSS poate fi utilizată totuși pentru afișarea documentelor XML structurate simple. XSL poate fi utilizat când sunt necesare capacități de formatare mai puternice sau pentru formatarea informațiilor cu structurare înaltă, cum sunt datele structurate XML sau documentele XML care conțin date structurate.

XSL și DSSSL

XSL este bazat pe o formă a limbajului DSSSL, numit DSSSL-o (*o* de la *online*) Limbajul DSSSL (Limbaj pentru stilul documentelor și specificația semantică – Document Style and Semantics Specification Language, standardul ISO/IEC 10179:1996) este bazat pe un dialect al limbajului de programare LISP, numit Scheme, care acceptă transformarea, interogarea și procesarea documentelor SGML. Partea de transformare din DSSSL nu este acceptată în XSL, iar partea cu foile de stil a fost redusă la minimul necesar pentru afișarea și tipărirea documentelor XML (și HTML).

XSL este compatibil cu principiile fundamentale de proiectare și modelul de procesare ale DSSSL. Totuși, așa cum dezvoltarea XML a scos în evidență „îmbunătățirile" care pot fi

făcute SGML-ului (limbajul din care provine), dezvoltarea XSL a identificat caracteristicile care pot fi puse în practică ale standardului DSSSL actual. O dată cu dezvoltarea specificațiilor pentru XSL, a fost anunțată și o propunere pentru un amendament la DSSSL. Intenția acestei propuneri este să se modifice standardul DSSSL pentru a se „legaliza" anumite caracteristici ale XSL, cum ar fi de exemplu extensiile CSS/HTML. În momentul în care acest amendament va fi introdus în standardul DSSSL și dacă acest lucru se va produce, DSSSL va deveni un superset al XSL.

După amendament, standardul DSSSL ar trebui să fie compatibil cu standardul DSSSL curent. Standardul modificat va avea trei lucruri în plus:

- sintaxă alternativă, compatibilă cu sintaxa XSL (XML).
- Extensii la limbajul de construire a arborelui obiectelor de curs pentru a accepta facilități ale XSL care nu sunt incluse în DSSSL; aceste caracteristici ar fi disponibile atât pentru sintaxa existentă, cât și pentru cea alternativă.
- Clase de obiecte de curs (HTML) noi și caracteristici noi pentru a ajuta formatarea paginilor Web.

NOTĂ

XSL este încă într-o fază incipientă de dezvoltare; în momentul în care acest capitol este scris, limbajul există numai sub forma unei propuneri care a fost înaintată consorțiului World Wide Web (W3C) pentru analiză.

Datorită caracterului de noutate și deci lipsei materialului documentar, multe din exemplele de cod folosite în acest capitol au la bază materialul cu propunerile pentru XSL. Pentru a exemplifica domeniile de utilizare potențiale ale XSL, am adăugat unele analize și comentarii.

Cum lucrează XSL

XSL este în principal un mecanism de stil condus prin date, care permite informației de formatare să fie asociată cu elementele din documentul sursă, pentru a rezulta produsul final formatat.

O foaie de stil XSL descrie procesul de formatare prin propriul set de elemente XML, numite *elementele documentului* XSL. Procesarea unui document XML e determinată de parcurgerea structurii sale și fuzionarea acesteia cu *specificația de formatare* rezultată din foile de stil active. Aceste instrucțiuni sunt apoi utilizate pentru a crea *obiecte de curs* (flow objects), care determină codarea aplicată documentului. Procesul de fuziune produce o structură arbore a obiectelor de curs numit *arborele obiectelor de curs*. Rezultatul formatat de la ieșire – care va fi tipărit sau afișat – este creat prin formatarea acestui arbore al obiectelor de curs

Obiectele de curs

Obiectele de curs, care descriu machetarea documentului, reprezintă componentele de machetare cum ar fi succesiunea paginilor, paragrafele, tabelele și obiectele grafice. Ele au proprietăți cum ar fi marginile, mărimea fonturilor, dimensiunile pe verticală și pe orizontală.

Demonstrând proveniența din DSSSL și aplicația în mediul HTML/CSS, XSL acceptă unele (dar nu toate) obiectele de curs folosite atât în DSSSL, cât și în HTML/CSS.

Principalele obiecte de curs din DSSSL

Obiectele de curs DSSSL sunt prin natură internaționale, astfel încât o singură foaie de stil poate fi folosită pentru a formata documente în limbaje naturale cu diferite direcții de scriere.

Nu toate obiectele de curs DSSSL sunt acceptate (pentru amănunte privind caracteristicile obiectelor de curs, consultați specificațiile DSSSL-o), dar următoarele sunt:

- **scroll:** Clasa obiectului de curs `scroll` e folosită ca obiect de curs la nivelul cel mai înalt pentru afișarea on-line care nu împarte în pagini rezultatul prelucrării. Aceasta acceptă obiectele de curs afișate. Mediul de vizualizare determină mărimea obiectului de curs pe direcție perpendiculară direcției de umplere.
- **paragraph:** Clasa obiectului de curs `paragraph` reprezintă un paragraf. Conținutul poate fi inline sau afișat. Obiectele de curs inline sunt formatate pentru a produce suprafețe cu linii. Obiectele de curs afișate specifică în mod implicit întreruperea (break), iar suprafețele acestora sunt adăugate la succesiunea finală a suprafețelor.
- **paragraph-break:** Folosit împreună cu clasa obiectelor de curs `paragraph-break`, un obiect de curs `paragraph` poate reprezenta un șir de paragrafe. Paragrafele sunt separate de obiecte de curs `paragraph-break`, care sunt indivizibile (atomice). Obiectele de curs `paragraph-break` sunt permise numai în cadrul obiectelor de curs `paragraph`. Toate caracteristicile care se aplică obiectului de curs `paragraph` sunt aplicabile și obiectului de curs `paragraph-break`. Caracteristicile obiectului de curs `paragraph-break` determină formatul pentru porțiunea de conținut corespunzătoare obiectului de curs `paragraph`, care urmează după obiectul de curs `paragraph-break` până la următorul obiect de curs `paragraph-break`, dacă există.
- **character:** Clasa obiectului de curs `character` este formatată pentru a produce o singură suprafață inline (obiectele de curs `character` pot fi numai inline). Suprafața poate fuziona cu alte suprafețe inline alăturate.
- **line-field:** Clasa obiectului de curs `line-field` este inline și are conținut inline. El produce o singură suprafață inline. Lățimea acestei suprafețe este egală cu valoarea caracteristicii `field-with` (lățimea câmpului). Dacă conținutul suprafeței `line-field` nu încape în lățime, suprafața crește până ce conținutul încape. Dacă `line-field` apare într-un paragraf, există o întrerupere după `line-field`. Puteți folosi aceste obiecte pentru a crea prefixe cu lățime fixă pentru elementele unei liste, astfel încât primul caracter de după prefix să înceapă în același loc, indiferent de mărimea prefixului.
- **external-graphic:** Clasa obiectului de curs `external-graphic` e folosită pentru grafica din entitățile externe. Obiectele de curs din această clasă pot fi inline sau afișate.
- **horizontal-rule și vertical-rule:** Aceste obiecte de curs reprezintă liniile despărțitoare orizontale și verticale; acestea pot fi inline sau afișate.
- **score:** Clasa obiectului de curs `score` e folosită pentru subliniere sau tăierea caracterelor cu o linie.
- **embedded-text:** Clasa obiectelor de curs `embedded-text` e folosită pentru înglobarea textului de la dreapta la stânga în text de la stânga la dreapta sau invers. Obiectele de curs din această clasă pot fi numai inline.
- **box:** Clasa obiectului de curs `box` e folosită pentru a crea o casetă în jurul unui șir de obiecte de curs; acest obiect de curs poate fi afișat sau inline. Dacă este afișat, acceptă orice fel de obiect de curs afișat. Dacă este inline, acceptă orice obiect de curs inline. Un obiect de curs `box` poate fi situat pe mai multe suprafețe, în care caz chenarul casetei adiacent întreruperii poate lipsi. Când caseta este in-line, chenarul este perpendicular pe direcția de scriere, iar când este afișată, chenarul e paralel cu direcția de scriere. Când caseta e afișată, mărimea ei (distanța dintre limitele chenarului) pe direcția scrisului este egală cu mărimea afișată a casetei, mai puțin indentările da la început și sfârșit. Mărimea de afișare pentru conținutul casetei este egală cu mărimea ei.

- **table:** Un obiect de curs `table` conține fie toate obiectele de curs din clasa `table-part`, fie cele din clasa `tabel-column`, `table-row` sau `table-cell`. Dacă conține obiecte de curs din clasa `table-column`, acestea trebuie să apară înainte de oricare obiecte de curs din alte clase. Un obiect de curs `table` poate fi doar afișat. Un tabel are două direcții asociate: direcția de avans pe rânduri și cea pe coloane.
- **table-part:** Clasa de obiecte de curs `table-part` este permisă numai în cadrul unui obiect de curs `table`. Un obiect de curs `table-part` este constituit din corpul tabelului, antetul și subsolul tabelului. În corpul tabelului sunt permise numai obiectele de curs `table-column`. Obiectele `table-row` și `table-cell` pot fi folosite oriunde în cadrul obiectului de curs `table-part`.

Rezultatul formatării obiectului de curs `table-part` este o succesiune de suprafețe. Fiecare suprafață e formată din conținutul antetului (cu excepția cazului în care e în mod explicit omis), urmat de conținutul corpului tabelului și de conținutul subsolului tabelului (cu excepția cazului în care e în mod explicit omis). Fiecare rând din corpul tabelului apare doar o singură dată și e păstrată ordinea rândurilor. Rândurile din antetul și subsolul tabelului sunt reproduse pentru fiecare suprafață rezultantă.

- **table-column:** Clasa obiectului de curs `table-column` reprezintă un obiect de curs atomic, care precizează caracteristicile aplicabile celulelor tabelului din aceeași coloană și același interval (span).
- **table-row:** Clasa obiectelor de curs `table-row` grupează celulele tabelului pe rânduri; toate celulele dintr-un `table-row` încep din același rând geometric. Obiectul de curs `table-row` acceptă orice obiect de curs din clasa `table-cell`. Obiectul de curs `table-row` poate apărea numai sub formă de copil al obiectelor `table-part` sau `table`.
- **table-cell:** Clasa obiectului de curs `table-cell` acceptă orice obiect de curs care poate fi afișat. Un obiect de curs `table-cell` poate apărea numai în postura de copil al obiectului de curs `table-row`, `table-part` sau `table`. Lățimea tabelului este egală cu suma lățimii celulelor.
- **table-border:** Obiectul de curs `table-border` este folosit pentru a preciza chenarul unei celule a tabelului sau al întregului tabel. Totuși, obiectul de curs `table-border` nu este permis în conținutul oricărui obiect de curs. Grosimea chenarului nu modifică lățimea celulelor, așezarea conținutului celulelor, lățimea tabelului sau mărimea suprafeței pe care o ocupă tabelul.
- **sequence:** Clasa obiectului de curs `sequence` e folosită pentru a concatena suprafețele produse de către fiecare copil, care pot fi inline sau afișate. Obiectul de curs `sequence` e util pentru precizarea caracteristicilor moștenite. Un alt obiect de curs acceptă un obiect de curs `sequence` numai dacă ar accepta fiecare dintre obiectele de curs din acea secvență.
- **display-group:** Clasa obiectelor de curs `display-group` concatenează alte obiecte de curs într-un mod similar cu obiectele `sequence`, dar creează și o nouă suprafață de afișare prin faptul că pornește de pe o linie nouă, care e urmată de o altă linie nouă (chiar dacă nu are conținut). Aceste grupuri de curs sunt folosite pentru controlul poziției grupurilor de obiecte de curs afișate.
- **simple-page-sequence:** Clasa obiectului de curs `simple-page-sequence` e destinată să producă un șir de suprafețe ale paginilor. Acest obiect de curs acceptă orice obiecte de curs afișate, dar nu este permis să apară în conținutul altei clase de obiecte de curs.

Obiectul de curs `simple-page sequence` poate avea un antet și un subsol dintr-o singură linie, conținând un text care este constant, dacă nu este nevoie de numerotarea paginilor. Un document poate conține mai multe `simple-page sequence`. De exemplu, fiecare capitol

al unui document poate fi un `simple-page sequence` separat, ceea ce ar permite ca titlul capitolului să fie utilizat în linia de antet sau de subsol.

Pagina este plină de sus și până jos. Mărimea afișajului pentru conținutul simple-page sequence este lățimea paginii, fără marginile din stânga și dreapta.

- **link:** Clasa obiectului de curs `link` reprezintă o legătură hipertext, care poate fi parcursă interactiv, în mod obișnuit făcând clic pe suprafețele care reprezintă obiectul de curs și conținutul acestuia. Totuși, după cum ați văzut, în XML nu este neapărat necesar să faceți clic pe o legătură pentru a o parcurge. O legătură poate conține atât obiecte de curs inline cât și afișate. Obiectele de curs `link` pot fi imbricate. Dacă sunt imbricate, este efectivă legătura cea mai interioară.

Principalele obiecte de curs HTML/CSS

XSL acceptă toate posibilitățile CSS, asfel încât este posibilă o translație mecanică precisă din CSS în XML.

În plus față de obiectele de curs DSSSL este definit un set de bază cu obiectele de curs HTML:

A	FRAMESET	SCRIPT
AREA	HR	SELECT
BASE	HTML	SPAN
BODY	IMG	TABLE
BR	INPUT	TBODY
CAPTION	MAP	TD
COL	META	TEXTAREA
COLGROUP	OBJECT	TFOOT
DIV	PARAM	THEAD
FORM	PRE	TR

Cu aceste obiecte de curs HTML/CSS, puteți folosi limbajul XSL cu HTML și CSS.

Caracteristicile obiectelor de curs

XSL este destul de ușor, astfel încât este posibil să se construiască foi de stil de bază pentru definițiile DTD simple doar prin învățarea numelor câtorva obiecte de curs și listarea mapării acestora pentru a tipări cu ajutorul regulilor de construcție.

Un obiect de curs este în realitate doar un obiect de formatare – o componentă tipografică cum este paragraful, tabela, o suprafață, ecuațiile matematice ș.a.m.d. Obiectele de curs au caracteristici și multe din numele acestor caracteristici sunt destul de evidente:

- **font-family-name:** Numele unei familii de fonturi.
- **font-size:** Un număr reprezentând mărimea fontului, precizat de obicei în puncte („pt"), cum ar fi de exemplu `10pt`. Îl mai puteți preciza în milimetri, țoli, pica sau alte unități.
- **line-spacing:** O cantitate care reprezintă mărimea spațiului dintre limita de jos a unei linii și cealaltă linie.
- **space-before:** Spațiul dinaintea unei suprafețe, cum ar fi spațiul introdus înaintea unui paragraf.
- **start-after:** Spațiul situat după o suprafață, cum ar fi cel introdus după un paragraf. Spațiul real dintre obiecte este suma dintre valoarea precedentă `space-after` și următoarea valoare `space-before`.
- **start-indent:** Indentarea, ca în cazul spațiului dintre marginea din stânga a textului și marginea din stânga a paginii la modurile de scriere de la stânga la dreapta (în limba engleză, de exemplu).
- **quadding:** Procesoarele de text denumesc acest termen „aliniere".

Toate obiectele de curs DSSSL au caracteristici specifice; pentru amănunte, ar trebui mai întâi să consultați specificația DSSSL-o. Totuși, multe dintre obiectele de curs mai sofisticate (cum sunt obiectele de curs `column-set` și `page-sequence`) au fost abandonate în limbajul XSL, ceea ce înseamnă că unele dintre caracteristicile descrise în specificațiile DSSSL-o nu sunt acceptate în XSL.

Caracteristicile utilizate de către obiectele de curs HTML/CSS sunt reprezentate de întregul set de proprietăți CSS, plus atributele HTML/CSS originale pentru cazurile în care setul de proprietăți CSS nu furnizează funcția respectivă.

Regulile de construcție

Fiecare limbaj pentru foile de stil constă dintr-o serie de declarații, care transformă elementele structurale în obiecte de formatare. Chiar și meniul Style din Microsoft Word face o astfel de mapare. Acesta ia elemente paragraf și le mapează la fonturi, culori și alte efecte tipografice.

În XSL, sintaxa pentru setarea acestei mapări e numită *regulă de construcție*, deoarece „construiește" un document formatat dintr-un document XML. Regulile de construcție conțin un *model* pentru a identifica elementele specifice din arborele sursă și o *acțiune* pentru a preciza un subarbore rezultant de obiecte de curs. Procesorul foii de stil procesează în mod recursiv elementele, pentru a genera un arbore complet al obiectelor de curs.

În plus față de regulile de construcție, XSL acceptă de asemenea *regulile de stil*, care permit fuziunea caracteristicilor. Pentru un anume element sursă poate fi invocată numai o singură regulă de construcție, dar *toate* regulile de stil. Procedând în acest fel, permiteți fuziunea caracteristicilor în XML, ca și cum ați utiliza foile de stil CSS în HTML.

Următorul exemplu prezintă o regulă de construcție pentru listele simple, folosind principalele obiectele de curs HTML/CSS. Prima regulă furnizează marginile și indentarea pentru întreaga listă; cea de-a doua furnizează marginile pentru articolele listei individuale. Indentarea negativă și întinderea (`SPAN`) pe lățime fixă a unui articol din listă, permite ca numărul articolului să fie generat și plasat la stânga fiecăruia.

```
<rule>
  <target-element type="list"/>
  <DIV margin-left="36pt" margin-top="12pt">
    <children/>
  </DIV>
</rule>

<rule>
  <element type="list">
    <attribute name="type" value="enum"/>
    <target-element type="item"/>
  </element>
  <DIV margin-top="4pt"
      margin-bottom="4pt"
      text-indent="-24pt">
  <SPAN width="24pt" height="12pt">
    <eval>
      formatNumber (childNumber(element),"1") + "."
    </eval>
  </SPAN>
  <children/>
```

```
  <DIV>
</rule>
```

Elementul fără conținut `children` este o instrucțiune către procesorul XSL pentru a procesa în mod recursiv copii elementului.

Următorul exemplu conține o regulă `root`, care creează obiectele de curs de cel mai înalt nivel `HTML` și `BODY`. Aceasta e urmată de o regulă care creează o diviziune (`DIV`), cu anumite caracteristici de stil pentru fiecare element `orders` și o regulă de stil (`style-rule`) care aplică stilul cu caractere aldine (`bold`) tuturor elementelor `customer`.

```
<xsl>
  <rule>
    <root/>

    <HTML>
      <BODY>
        <children/>
      </BODY>
    </HTML>
  </rule>

  <rule>
    <target-element type="orders"/>
    <DIV font-size="14pt" font-family="serif">
      <children/>
    </DIV>
  </rule>

  <style-rule>
    <target-element type="customer"/>
    <apply font-weight="bold"/>
  </style rule>
</xsl>
```

Regula e precizată cu ajutorul elementului `rule`, care conține un model de identificare a elementului asupra căruia se aplică regula. Regula conține și o acțiune care specifică obiectele de curs care se construiesc.

De exemplu, să presupunem că un document XML conține :

```
<P>This is a truly <emph>momentous</emph> occasion.</P>
```

Modelul din următoarea regulă XSL identifică elementele de tip `emph` și definește o acțiune care produce un obiect de curs `SPAN` cu caractere aldine:

```
<rule>
  <!-- modelul -->
  <target-element type="emph"/>
    <!-- actiunea -->
  <SPAN font-weight="bold"/>
    <children/>
  </SPAN>
</rule>
```

O singură acțiune poate fi asociată cu mai multe modele. Următorul exemplu construiește `SPAN` pentru modelele elementelor `emph` și `strong`:

```
<rule>
    <target-element type="emph"/>
```



```
    <target-element type="strong"/>
    <SPAN font-weight="bold"/>
      <children/>
    </SPAN>
</rule>
```

Modele

Partea cu modele a regulii de construcție trebuie să conțină un element `target-element` care indică elementul asupra căruia se aplică regula. Modelul vă permite de asemenea să identificați elementele aplicabile după contextul acestora. Sintaxa e bazată pe sintaxa pointerului extins din XML. Puteți selecta elementele după cum urmează:

Părinții acestora

Copii acestora

Folosind elemente de înlocuire (wildcards) în ierarhia părinte/copil (pentru a selecta în mod arbitrar părinții/copiii de la orice nivel în jos sau în sus)

Atributele acestora

Poziția acestora în comparație cu elementele frate-soră (de exemplu, al treilea copil)

Unicitatea acestora în relația cu elementele frate-soră

Strămoșii elementului pot fi reprezentați în model prin oglindirea ierarhiei documentului. Elementul are atribute identice cu elementul țintă `target-element`, dar elementul definit de atributul său `type` este parte a contextului elementului aplicabil în document, nu al elementului căruia i se va aplica regula.

Imbricarea elementelor `elements` din model denotă contextul relației acestora cu elementul `target-element` din sursă. De exemplu, următorul model face corespondența dintre elementele `title` care au ca părinte elementul `section` și ca bunic elementul `chapter`:

```
<element type="chapter">
  <element type="section">
    <target-element type="title"/>
  </element>
</element>
```

În mod similar pot fi precizați și copiii unui element. Următorul model face corespondența dintre elementele sursă `table` care au un element copil `child`:

```
<target-element type="table">
  <element type="title"/>
</target-element>
```

În interiorul elementului `target-element` se pot introduce mai multe elemente `element`. Acest lucru indică numai faptul că aceste elemente trebuie să fie prezente, dar nu în orice ordine. Următorul model face corespondența dintre elementele `man` care conțin atât elementul `rich`, cât și elementul `old`:

```
<target-element type="man">
  <element type="rich"/>
  <element type="old"/>
</target-element>
```

Rădăcina arborelui documentului poate fi identificată în modelul special `root`:

```
<rule>
<root/>
    <HTML>
```

```
    <BODY>
      <children/>
    </BODY>
  </HTML>
</rule>
```

Elementul părinte al unui element este elementul care-l conține, iar elementul conținut în el este elementul copil. Relația este directă, dar nu există nici un concept direct pentru bunic sau nepot. O relație bunic trebuie să fie exprimată ca fiind părintele părintelui, iar o relație nepot trebuie să fie exprimată ca fiind copilul copilului. Pentru a rezolva acest lucru, în cadrul modelului pot fi folosite elementele de înlocuire, iar elementul `any` poate fi folosit pentru selectarea oricărui element din ierarhia elementelor.

Următorul exemplu aplică cele de mai sus tuturor elementelor `para` care au un strămoș `ancestor` și un urmaș `emph`:

```
<element type="appendix">
  <any>
    <target-element type="para">
      <any>
        <element type="changed"/>
      </any>
    </target-element>
  </any>
</element>
```

Ambele elemente, `target-element` și `element` pot fi folosite drept elemente de înlocuire, omițând atributul `type`. Aceste elemente coincid apoi cu unul din elementele din ierarhia elementelor. Următorul model face corespondența dintre elementele care au un bunic de tip `data-samples`, indiferent de tipul elementului `target-element` sau elementul părinte:

```
<element type="data-samples">
  <element>
    <target-element/>
  </element>
</element>
```

Atribute

Atributele unui element, sau ale oricărui element strămoș sau urmaș al lui, pot fi și ele folosite pentru a verifica dacă se aplică sau nu o anumită regulă de construcție.

Pentru selectarea unui element cu ajutorul unui anumit atribut și al unei anumite valori, se include atributul cu valoarea acestuia într-un element `element` sau `target-element`.

Următorul exemplu face corespondența dintre elementul `item` care este un copil al elementului `list`, a cărui atribut `type` are valoarea `enum`:

```
<element type="list">
  <attribute name="type" value="enum"/>
  <target-element type="item"/>
</element>
```

De asemenea, puteți testa dacă un atribut există. Când atributul `has-value` are valoarea `"yes"`, elementul corespunde modelului dacă atributului respectiv i s-a dat o valoare (fie printr-o declarație explicită, fie prin primirea unei valori prestabilite). Când atributul `has-value` are valoarea `"no"`, elementul corespunde modelului dacă valoarea predefinită a atributului este `#IMPLIED` și nu a fost atribuită în mod explicit. Dacă valoarea `value` a fost atribuită, atributul `has-value` e omis.

Următorul exemplu face corespondența dintre elementul `item`, care este copilul elementului `list` al cărui atribut `compact` are o valoare oarecare (caz contrar celui în care nu este atribuit și este ca urmare implicat):

```
<element type="list">
  <attribute name="compact" has-value="yes"/>
  <target-element type="item"/>
</element>
```

Un singur element `target-element` sau `element` acceptă mai multe elemente `attribute`, după cum se vede mai jos:

```
<target-element type="reference">
  <attribute type="href" has-value="yes"/>
  <attribute type="used-as" value="direct-quote"/>
</target-element>
```

În mod asemănător foilor CSS, XSL tratează atributele `ID` și `CLASS` într-un mod special. Atributul unui `element` este un atribut `ID` dacă este declarat în acest fel în definiția DTD din limbajul XML. Deoarece și limbajul XSL trebuie să lucreze cu documente XML care nu au DTD, foile de stil pot specifica atributele `element` care trebuie tratate ca atribute `ID`. Acest lucru se face prin folosirea unui element `ID`, cu atributul `attribute` care precizează că numele atributului trebuie tratat ca un atribut `ID`. Următorul exemplu arată că atributul `name` al oricărui element trebuie tratat ca fiind un atribut de tip `ID`:

```
<id attribute="name"/>
```

Elementul `target-element` sau `element` poate preciza un atribut `ID` pentru a selecta un element care are atributul `ID` cu aceeași valoare.

În același mod, o foaie de stil poate utiliza elementul `class` pentru a declara numele unui atribut care ar trebui să fie tratat drept atributul `class`. Următorul exemplu creează un atribut `family` al oricărui element tratat ca fiind un atribut `class`:

```
<class attribute="family"/>
```

Elementul `target-element` sau `element` poate preciza un atribut `class` pentru a selecta un element care are atributul `class` cu aceeași valoare.

Calificative

În mod asemănător selectării elementelor XML în funcție de poziția acestora în arborele documentului, puteți îngusta baza de selecție prin alegerea elementelor care au anumite poziții sau proprietăți prin folosirea calificativelor cu modele.

Iată două calificative, `position` și `only`; fiecare este exprimat ca un atribut al elementelor `element` sau `target-element`.

Calificativul `position` specifică poziția unui element printre frați; acesta are următoarele valori:

- **first-of-type**: Elementul trebuie să fie primul frate de același tip.
- **last-of type**: Elementul trebuie să fie ultimul frate de același tip.
- **first-of-any**: Elementul trebuie să fie primul frate de oricare tip.
- **last-of-any** :Elementul trebuie să fie ultimul frate de oricare tip.

Calificativul `only` precizează dacă elementul sursă are frați; valorile sale sunt următoarele:

- **of-type**: Elementul trebuie să nu aibă ca frați elemente de același tip.
- **of-any**: Elementul trebuie să nu aibă frați deloc.

Următorul model corespunde primului element din listă:

```
<element type="list">
  <target-element type="item" position="first-of-type"/>
</element>
```

Următorul model corespunde singurului element din listă:

```
<element type="list">
  <target-element type="item" only="of-type"/>
</element>
```

Prioritatea

Este posibil ca un element să corespundă modelelor din punctul de vedere al mai multor reguli diferite. Când corespunde mai multor modele, atunci e folosit modelul cu *specificitatea cea mai mare*, conform ordinii următoare:

1. Modelul cu cea mai mare importanță; importanța unui model este valoarea atributului `importance` al regulii care-l conține.
2. Modelul cu cele mai multe atribute `ID`
3. Modelul cu cele mai multe atribute `class`.
4. Modelul cu cele mai multe elemente `element` sau `target-element` care au și un atribut `type`.
5. Modelul cu mai puține elemente de înlocuire (element de înlocuire este un element `any`, sau un element `element` sau `target-element` care nu are un atribut `type`).
6. Modelul cu prioritatea specificată cea mai înaltă; prioritatea unui model este dată de valoarea atributului `priority` corespunzător regulei care-l conține.
7. Modelul cu cele mai multe calificative `only`.
8. Modelul cu cele mai multe calificative `position`.
9. Modelul cu cele mai multe specificații ale atributelor.

Pentru a determina care dintre cele două modele are specificitatea cea mai mare, se aplică criteriul precedent în ordinea arătată până când se identifică exact *unul* dintre modele.

Întotdeauna trebuie să existe numai un model cu specificitatea mai mare decât a celorlalte modele care corespund.

Acțiuni

Când regula care se aplică unui element a fost identificată, este invocată partea de *acțiune* a regulei. Structura elementelor care vor compune acțiunea reprezintă structura obiectelor de curs care vor fi create.

Elementele obiecte de curs pot fi combinate cu text (care generează obiecte de curs `caracter`) și text generat cu ajutorul codului (elementul `eval`) pentru a crea subarborii obiectelor de curs. Următorul exemplu selectează elementele `chapter`, pune deasupra și dedesubtul fiecăruia o linie orizontală și adaugă prefixul `"Chapter"`, urmat de numărul capitolului și două puncte:

```
<rule>
  <!-- model -->
  <target-element type="chapter"/>
  <!-- actiunea -->
  <sequence>
    <horizontal-rule/>
    <paragraph space-before="2pt">
```



```
      Chapter
        <eval>formatNumber(element.childNumber(),'001')
        </eval>
      :
      <children/>
    </paragraph>
    <horizontal-rule/>
  </sequence>
</rule>
```

Aplicarea caracteristicilor

În cadrul regulilor de formatare, în paralel cu specificarea acțiunilor de formatare puteți stabili caracteristicile obiectelor de curs, prin aplicarea acestora ca atribute ale elementelor care reprezintă obiectul de curs.

Pentru principalele obiecte de curs HTML, proprietățile CSS pot fi aplicate în mod direct ca atribute, fără a se folosi atributul HTML `style`, după cum se arată în exemplul următor:

```
<rule>
  <target-element type="para"/>
  <DIV font-family="Courier"
       font-size="12pt"
       font-style="normal"
       text-align="justify"
       margin-top="10pt"
       margin-bottom="6pt">
      <children/>
  </DIV>
</rule>
```

Selecția în cadrul unei acțiuni

Selecția din cadrul unei acțiuni controlează care dintre elemente sunt procesate după ce o regulă a identificat un element. Selecția folosește două elemente: elementul `children` și elementul `select` combinat cu atributul `from`.

Elementul `children` procesează copiii imediat următori ai unui element. Următorul exemplu creează un obiect de curs `display-group` pentru elementul `chapter` și apoi procesează elementele copil imediat următoare:

```
<rule>
  <target-element type="chapter"/>
  <display-group>
    <children/>
  </display-group>
</rule>
```

Elementul `select` procesează copiii sau moștenitorii elementului și asigură o filtrare puternică cu ajutorul modelelor. Exemplul anterior poate fi rescris folosind elementul `select` cu atributul `from` stabilit la `children`, pentru a se obține același efect:

```
<rule>
  <target-element type:"chapter"/>
  <display-group>
    <select from="children"/>
  </display-group>
</rule>
```

Cele două valori ale atributului `from` sunt:

- **`children`:** Sunt procesați copiii elementului.
- **`descendants`:** Sunt procesați urmașii elementului.

Elementul `select` procesează în mod prestabilit numai copiii imediat următori ai elementului. Pentru a procesa toți urmașii elementului sursă (elementele copii, copiii acestora, copiii copiilor acestora ș.a.m.d.) atributul `from` trebuie setat pe poziția `descendants`. Următorul exemplu procesează toate elementele `title` din elementul `chapter`:

```
<rule>
  <target-element type="chapter"/>
  <paragraph>
    <select from="descendants">
      <target-element type="title"/>
    </select>
  </paragraph>
</rule>
```

Filtre

Pot fi adăugate filtre elementului `select`, astfel încât să fie procesate numai elementele care corespund cu modelul filtrului. Modelul filtrului folosește aceeași sintaxă ca partea de model a regulei. În exemplul următor, după ce se creează un paragraf pentru elementul `man`, numai elementele copil `rich` sunt selectate pentru procesare:

```
<rule>
  <target-element type="man"/>
  <paragraph>
    <select>
      <target-element type="rich"/>
    </select>
  </paragraph>
</rule>
```

Pentru a efectua reordonarea simplă pot fi folosite într-o singură acțiune mai multe elemente `children` sau `select`. Următorul exemplu separă elementele `calls` în două tabele; primul tabel e completat cu apelurile locale, iar cel de-al doilea tabel e completat cu apelurile internaționale:

```
<rule>
  <target-element type="telephone"/>
  <display-group>
    <table>
      <select from="descendants">
        <element type="calls">
          <target-element type="local"/>
        </element>
      </select>
    </table>
    <table>
      <select from="descendants">
        <element type="calls">
          <target-element type="foreign"/>
        </element>
      </select>
    </table>
  </display-group>
</rule>
```


În mod prestabilit, elementele `children` și `select` procesează elementele copil ale elementului care corespunde modelului regulei.

Atributul `ancestor` este folosit pentru a procesa elementele copil în legătură cu un element superior din document. Exemplul următor identifică departamentul unui angajat și apoi procesează elementul copil `group` al departamentului:

```
<rule>
  <target-element type="employee"/>
  <paragraph>
    <select ancestor="department"/>
      <target-element type="group"/>
    </select>
  </paragraph>
</rule>
```

Macroinstrucțiuni cu obiecte de curs

Acțiunile pot fi separate în afara regulilor și salvate ca *macroinstrucțiuni*, care vă permit adunarea laolaltă a obiectelor de curs și referirea la acestea ca și cum ar fi un singur obiect de curs.

În acest exemplu, se definește o macroinstrucțiune pentru un paragraf încasetat conținând cuvântul `Note:`, care precede conținutul. Macroinstrucțiunea e apelată dintr-o regulă pentru elementele `note`:

```
<define-macro name="note">
  <box>
    <paragraph>
      Note:
      <contents/>
    </paragraph>
  </box>
</define-macro>

<rule>
  <target-element type="note"/>
  <note-para font-size="14pt">
    <children/>
  </note-para>
</rule>
```

Elementul `contents` din macroinstrucțiune se referă la conținutul referinței macroinstrucțiunii (în acest caz, elementul `children`).

Argumentele și valorile prestabilite ale argumentelor macroinstrucțiunii sunt precizate cu ajutorul elementului `arg`. Argumentele pot fi accesate în interiorul macroinstrucțiunii cu sintaxa `"=" + argument name`. Exemplul următor definește o macroinstrucțiune pentru un `list-item` cu argumente care controlează indentarea și șirul folosit ca marcă:

```
<define-macro name="list-item">
  <arg name="marker" default="•"/>
  <arg name="indent" default="0.25in"/>
  <paragraph
    first-line-start-indent="=-indent"
    start-indent="=indent">
```

```
  <line-field field-width="=indent">
    <eval>marker</eval>
  </line-field>
  <contents/>
</paragraph>
</define macro>

<rule>
  <target-element type="item"/>
  <list-item marker="+">
    <children/>
  </list-item>
</rule>
```

Regula de construcție prestabilită

Limbajul XSL are încorporată o regulă de construcție prestabilită care permite continuarea procesării recursive în cazurile în care nu poate fi găsit modelul corespunzător. Puteți anula regula prestabilită precizând următoarele:

```
<rule>
  <target-element/>
  <children/>
</rule>
```

Reguli de stil

Un element XSL poate fi asociat cu caracteristicile obiectelor de curs prin *reguli de stil*. În mod asemănător regulilor de construcție, regulile de stil sunt compuse dintr-un model și o acțiune. Modelul selectează elementul, iar acțiunea precizează caracteristicile obiectului de curs.

Exemplul următor prezintă o regulă de stil simplă care transformă toate elementele `strong` în aldine:

```
<style-rule>
  <target-element type="strong"/>
  <apply font-weight="bold"/>
</style-rule>
```

Regulile de stil acceptă caracteristica CSS prin care mai multe reguli ale foilor de stil pot contribui cu proprietăți (caracteristici) la prezentarea unui element. De exemplu, următoarele reguli CSS sunt destinate să transforme tipul fonturilor `font-weight` al elementelor `strong` în elemente tip `H1 bold`, iar atributele `color` în roșu (`red`):

```
h1 {font-weight:bold; color:blue}
h1 strong {color:red}
```

Rescrisă în limbajul XSL, declarația arată astfel:

```
<style-rule>
  <target-element type="h1"/>
  <apply font-weight="bold" color="blue"/>
</style-rule>

<style-rule>
  <element type="h1">
    <any>
      <target-element type="strong"/>
    </any>
```



```
  </element>
  <apply color="red"/>
</style-rule>
```

NOTĂ

Acest exemplu este puțin incorect deoarece în linia 12 a codului XSL e necesar să se reproducă doar două linii ale codului CSS. Sintaxa XSL pentru selectare și formatare îl fac pe acesta mult mai puternic decât CSS, dar după cum demonstrează acest exemplu simplu, dacă cerințele dumneavoastră sunt simple, foile CSS pot fi preferate chiar limbajului XSL.

Numele stilurilor

În loc să precizeze stilul pentru fiecare element sau set de elemente pe care le selectați, puteți pune nume stilurilor și să le utilizați când doriți.

Puteți pune etichetă unui set de caracteristici prin folosirea atributului `name`. Următorul exemplu definește un stil numit `title-style`:

```
<define-style name="title-style"
              font size="18pt"
              font-weight="bold"/>
```

Stilurile denumite în acest mod pot fi apelate de un element obiect de curs prin precizarea numelui stilului ca valoare a atributului `use`. Următorul exemplu aplică stilul `title-style`, definit în exemplul precedent, elementelor `heading` din elementul `chapter`:

```
< rule>
  < element type="chapter">
    <target-element type="heading"/>
  </element>
  <paragraph use="title-style">
    <children/>
  </paragraph>
</rule>
```

Regula de stil poate folosi un element cu nume de stil în loc de acțiunea `apply`, lucru care vă permite să adăugați caracteristici din regula de stil unui stil care are nume.

Moduri

Este nevoie uneori ca elementele limbajului XML să fie afișate în mai multe locuri. De exemplu, s-ar putea să fie necesar ca elementul `title` să fie afișat atât în cuprins, cât și în conținut. În fiecare loc, s-ar putea ca elementele să fie formatate diferit.

Dacă doriți să procesați un subcapitol (sau pe toate) din documentul care folosește unele reguli de construcție diferite, XSL poate face acest lucru foarte ușor, folosind „modurile". În esență, *modul* este chiar o schemă de formatare care posedă nume.

Un element poate fi procesat în funcție de un mod care are nume. Unui mod care are nume i se asociază o regulă, folosind atributul `mode` al regulei.

Elementele `children` și `select` pot fi transformate într-un mod care are nume prin precizarea atributului `mode`:

```
<rule>
  <root/>
  <HTML>
    <BODY>
      <DIV name="table-of-contents">
```

```
      <children mode="toc-mode"/>
    </DIV>
    </children>
  </BODY>
 </HTML>
</rule>

<rule>
 <target-element type="title"/>
 <DIV font-size="14pt" font weight="bold">
   </children>
 </DIV>
</rule>

<rule mode="toc-mode">
 <target-element type="title"/>
 <DIV font-size="12pt">
   <children/>
 </DIV>
</rule>
```

Referința la o foaie de stil din limbajul XML

Grupul de lucru XML nu a definit încă mecanismul de asociere al documentelor XML cu foile de stil XSL. Până ce acest lucru va fi terminat și vor fi făcute și alte propuneri, ar trebui să folosiți instrucțiunile de procesare a foilor de stil XML pentru a putea să asociați o foaie de stil cu un document XML. De exemplu, introduceți o linie asemănătoare celei care urmează la începutul fișierului sursă XML (după declarația XML):

```
<?xml-stylesheet href="article.xsl" type="text/xsl" ?>
```

Importul unei foi de stil

O foaie de stil XSL poate importa alte foi de stil XSL folosind elementul `import`, cu atributul `href`:

```
<import href="article.xsl"/>
```

Folosirea unui element `import` are același efect ca și cum foaia de stil localizată de valoarea atributului `href` ar fi inclusă fizic în punctul de unde este apelată.

Stiluri inline

Dacă nu doriți să definiți complet stilul de formatare în foile de stil XSL, cum sunt de exemplu situațiile în care doriți să rezervați spațiu pentru flexibilitate, XSL vă permite să anulați caracteristicile de stil dintr-un document, prin includerea declarațiilor de stil în documentul propriu-zis (inline).

Atributele de formatare

Pentru a anula specificațiile foilor de stil XSL pentru un anumit element, adăugați atributele de stil elementului din documentul XML. Următoarea declarație, de exemplu, face ca elementul `para` să fie scris cu caractere aldine, indiferent de ce se specifică în foaia de stil XSL pentru celelate:

```
<para xsl::font-weight="bold">
```


Reguli de formatare

Pentru a anula specificațiile foilor de stil XSL pentru grupuri de elemente (într-un mod similar în care se folosesc clasele în CSS), introduceți o regulă XSL la începutul documentului XML. Pentru a aplica regula și a anula conținutul foii de stil, folosiți aceeași valoare a atributului `ID` pentru regula de stil și pentru elementele cărora li se aplică regula, după cum se arată în acest exemplu:

```
<rule>
 <target-element id="local"/>
 <DIV font-weight="bold">
   <children/>
 </DIV>
</rule>
...
<para id="local">
```

Anularea specificațiilor foii de stil

Cele două metode descrise anterior vă permit să anulați specificațiile foii de stil XSL prin introducerea instrucțiunilor de formatare în documentul XML. Privind din direcție opusă – adică din interiorul foii de stil XSL – puteți introduce în foaia de stil respectivă atributul `use` al unui element obiect de curs care are valoarea `#source` în regula pentru tipul elementului. Această metodă face ca orice atribute al căror nume coincide cu cel al unei caracteristici DSSSL moștenite să fie aplicată elementului din spațiul rezervat numelui XML (adică, în documentul XML):

```
<rule>
 <target-element type="para"/>
 <DIV use="#source">
   <children/>
 </DIV>
</rule>
```

Limbajul de scriptare XLS

Bazele limbajului de scriptare XLS sunt oferite de standardul ECMAScript. ECMAScript – sau mai exact „Standard ECMA-262 ECMAScript: A general purpose scripting language" – se bazează în principal pe JavaScript, produs de Netscape. Se spune că Netscape a încercat să transforme JavaScript într-un limbaj formal, prin acceptarea ca standard ISO. Se pare că acesta a fost respins inițial ca fiind inacceptabil în forma remisă, deoarece era incomplet.

Fondată în mod oficial în anul 1961, ECMA (cunoscută anterior ca Asociația Europeană a Producătorilor de Calculatoare – European Computer Manufacturers' Association) a fost rebotezată ca „European association for standardization information and communication systems" (Asociația europeană pentru standardizarea informației și sisteme de comunicații), în 1994. Din 1987, ECMA a devenit membru de legătură cu statut A al Comitetului tehnic mixt 1 ISO/IEC (JTC 1), care este responsabil și cu SGML.

Specificația ECMA-262 este un document extensiv, așa că nu face obiectul acestui capitol. Puteți obține specificația în mod gratuit de pe situl Web al ECMA, la adresa `http://www.ecma.ch/` în format Adobe Acrobat (PDF) sau Word. În ultima parte a acestui capitol mă voi axa pe modurile în care XSL extinde și folosește limbajul ECMAScript.

Limbajul de scriptare XLS extinde ECMAScript prin oferirea datelor de tip cantitate, care reprezintă lungimile și cantitățile provenite din lungimi, cum ar fi suprafețele și volumele. Unitatea de bază este metrul din Sistemul Internațional (m).

Sunt definite următoarele unități derivate:

- cm (0,01m)
- mm (0,001m)
- pc (1/6 in)
- pt (1/12 pc)

Cantitățile dependente de dispozitiv, cum ar fi pixelii, sunt utilizate în XSL folosind conceptul `length-spec`.

Propunerea XSL precizează faptul că va fi inclus un set de funcții încorporate și că acesta va reprezenta un subset al funcțiilor incluse în standardul DSSSL.

Scriptarea în XSL

Limbajul de scriptare XSL (ECMAScript) poate fi folosit în XSL în mai multe feluri:

- Poate fi folosit la nivelul cel mai de sus (în afara regulilor de construcție) pentru declararea variabilelor și definirea funcțiilor dintr-un element `define-script`:

```
<define-script>
  var defaultFontSize = "12pt";
   function hierarchicalIndent (elementType, element)
   {
   return length(hierarchicalNumberRecursive
                 (elementType, element)) * 12pt;
   }
</define script>
```

Funcția `hierarchicalNumberRecursive` este o funcție DSSSL care returnează o suprafață ce reprezintă numerele copil al tipului de element solicitat, la fiecare nivel ierarhic. Formatarea acestei liste produce un sistem de numerație.

- Dacă valoarea unui atribut care specifică o caracteristică începe cu semnul egal (=), restul atributului e interpretat ca o expresie ECMAScript:

```
<rule>
  <element type="list">
    <target-element type="item">
  </element>
  <DIV font-size="=defaultFontSize"
    <children/>
  </DIV>
</rule>
```

- În cadrul unei acțiuni, un element `eval` care conține o expresie ECMAScript poate fi folosită pentru calculul textului generat. Valoarea returnată de expresie este convertită într-un șir și apoi la o secvență de obiecte de curs `character`.

Elementul `eval` poate conține un și un șir de declarații ECMAScript. Valoarea returnată de declarația `return` e transformată într-un șir și apoi intr-o secvență de obiecte de curs `character`, după cum se vede:

```
<rule>
  <target-element type="title">
  <DIV>
    <eval>
       formatNumberList(
         hierarchicalNumberRecursive ("title", element),
         "",".")
    </eval>
    <children/>
  </DIV>
</rule>
```


Elementul `select` poate filtra o listă de noduri oarecare, specificate printr-un script în cadrul atributului `from`. Acest exemplu determină cel mai apropiat strămoș `department` și returnează copiii `manager` ai elementului:

```
<select from='=ancestor("department",element)'>
  <target-element type="manager"/>
</select>
```

O foaie de stil simplă cu script

Următoarea foaie de stil, care face parte din propunerea pentru limbajul XSL, reprezintă o traducere în XSL a unor părți ale foii de stil DSSSL pe care Jon Bosak a scris-o pentru citirea HTML.

Acest script prezintă modul în care construcții declarative pot fi combinate cu scriptul în liste de format:

```
<xsl>
  <define-script>
    var bfSize = 10pt;
    var paraSep = bfSize/2.0;
    var blockSep = paraSep*2.0;
    var bodyStartIndent = 6pc;
    var lineSpacingFactor = 1.1;

    function inlist(element)
    {
      return
        haveAncestor(element, "OL") |
        haveAncestor(element, "UL") |
        haveAncestor(element, "DIR") |
        haveAncestor(element, "MENU") |
        haveAncestor(element, "DL") |
    }
    function olstep(element)
    {
      var x = length(hierarchicalNumberRecursive(element, "OL")) % 4;
      if (x == 1) return 1.2*bfSize;
      if (x == 2) return 1.2*bfSize;
      if (x == 3) return 1.6*bfSize;
      if (x == 0) return 1.4*bfSize;
    }
  </define-script>

  <define-style name="p-style"
    font-size="=bfSize"
    line-spacing="=bfSize * lineSpacingFactor"/>

  <rule>
    <target-element type="OL"/>

    <display-group
       space-before="=inlist(element) ? paraSep : blockSep"
       space-after="=inlist(element) ? paraSep : blockSep"
```

```
      start-indent="=inlist(element) ?
      inheritedStartIndent(element) : bodyStartIndent" >
    <children/>
  </display-group>
</rule>

<rule>
<element-type="OL">
  <target-element type="LI"/>
</element>

<paragraph use="p-style"
    space-before='=attributeString
      (ancestor("OL"), "compact") ? 0 : paraSep'
      start-indent="= inheritedStartIndent(element) + olstep(element)"
      first-line-start-indent="=-olstep(element)" >
    <line-field field-width="=olstep(element)">
      (
      <eval>
        var child = childNumber(element);
        var x = hierarchicalNumberRecursive("OL") % 4;
        if (x == 1) return formatNumber(child, "1");
        if (x == 2) return formatNumber(child, "a");
        if (x == 3) return formatNumber(child, "i");
        if (x == 0) return formatNumber(child, "a");
      </eval>
      )
    </line-field>
    <children/>
  </paragraph>
</rule>

<rule>
  <element type="OL">
    <element type="LI">
      <target-element type="P"/>
    </element>
  </element>
  <paragraph use="p-style"
   start-indent="=inheritedStartIndent(element)
   + olstep(element)"first-line-start-indent=
     "=-olstep(element)" >
    <children/>
  </paragraph>
</rule>

</xsl>
```

Foi de stil cu structură fixă

Dacă doriți să creați cod XSL care este valid conform unei singure definiții DTD fixe în limbaj XML pentru toate foile de stil XSL (definiția DTD este este inclusă în propunerea XSL, care e prezentată în subcapitolul următor), există o variantă de sintaxă folosind stilurile care posedă nume:

```
<style-rule>
  <target-element type="title"/>
```



```
  <apply use="title-style" quadding="center"/>
</style-rule>
```

Există și o sintaxă alternativă pentru apelarea macroinstrucțiunii:

```
<rule>
  <target-element type="item"/>
  <invoke macro="list-item">
    <arg name="marker" value="+">
    <children/>
  </invoke>
</rule>
```

O definiție DTD din XML pentru reguli XSL

Următoarea definiție DTD XML e preluată integral din propunerea XSL. Folosind varianta de sintaxă pentru regula de stil și macroinstrucțiune, descrise în subcapitolul precedent, această definiție DTD XML poate fi folosită pentru a se asigura că foaia de stil XSL este corectă din punct de vedere al sintaxei XML:

```
<!-- Aceasta DTD are numai scop explicativ. Ea presupune ca
entitatea parametru actiune a fost definita ca un grup-sau
de elemente obiect de curs. Entitatea parametru de stil
este folosita pentru a reprezenta un grup-sau al elementului
apply si stiluri definite cu define-style. -->

<!ENTITY % pattern (root | (target-element | element | any)*)">

<!ELEMENT rule (%pattern;, %action;)>

<!ELEMENT style-rule (%pattern;, %style)>

<!ELEMENT root EMPTY>

<!ATTLIST (rule | style-rule)
     priority   NUMBER     #IMPLIED
     importance NUMBER     #IMPLIED
     mode       NAME       #IMPLIED>

<!ELEMENT target-element
     (attribute*, (element+ | any)?) -(target-element)>

<!ELEMENT element
     (attribute*, (target-element | element | any)?) >

<!ELEMENT any EMPTY >

<!ATTLIST (target-element | element)
     type       NAME           #IMPLIED
     id         NAME           #IMPLIED
     class      NAME           #IMPLIED
     only     (of-type|of-any)    #IMPLIED
     position   (first-of-type|last-of-type|
                 first-of-any|last-of-any) #IMPLIED
>

<!ELEMENT attribute EMPTY >

<!ATTLIST attribute
     name       NAME           #REQUIRED
     value    CDATA    #IMPLIED
     has-value    (yes|no) 'yes'
```

Rezumat

În acest capitol, ați învățat următoarele:

- Faptul că limbajul XML este destinat să fie o variantă mai puternică decât foile de stil CSS, care permite modalități sofisticate de selecție și formatare a elementelor
- Diferențele semnificative dintre XSL ca limbaj de stil pentru XML și pentru HTML și CSS ca limbaj de stil numai pentru HTML
- Diferențele majore dintre XML ca limbaj de stil pentru XML/HTML/SGML și DSSSL ca limbaj de stil și transformare pentru XML/HTML/SGML
- Sintaxa de bază a limbajului XSL și modul de construire a regulilor structurii XSL
- Cum să scriem o foaie de stil XSL și cum să o apelăm dintr-un document XML
- Cum se anulează specificațiile foii de stil XSL prin includerea definițiilor de stil inline în conținutul documentelor XML
- Principalele aspecte ale limbajului de scriptare XSL (ECMAScript)

Deși putem spune că limbajul DSSSL se află și el în perioada de copilărie (în ciuda faptului că au trecut mai mulți ani de la apariție, a devenit standard ISO abia în anul 1996), a fost deja abandonat pe scară largă, deoarece este pur și simplu prea dificil pentru publicul larg, cu excepția programatorilor, care-l folosesc într-adevăr. DSSSL-o (și predecesorul său cu viață foarte scurtă, DSSL-lite) a reprezentat o încercare de a adapta DSSSL la aplicațiile cotidiene. Se poate spune pe bună dreptate că DSSSL nu a depășit niciodată faza de proiectare.

Pe de altă parte, CSS reprezintă un instrument foarte util pentru formatarea paginilor HTML, dar este mult prea limitat pentru aplicațiile mai complicate, care necesită un model de editare mai ușor de recunoscut.

Deși are rădăcini adânci în DSSSL, limbajul XML se deosebește în mod esențial față de acesta. XML încearcă să folosească puterea și tăria limbajului DSSSL, preluând în același timp ușurința și simplitatea foilor de stil CSS. În această privință, XSL este corespondentul perfect al XML, care la rândul lui, încearcă să folosească avantajul tăriei și puterii SGML, păstrând ușurința și simplitatea acestuia.

O trăsătură care poate fi factorul decisiv în succesul limbajului XSL este aceea că el poate fi exprimat chiar în XML. Același editor pe care-l folosiți pentru a crea documentele XML poate fi la fel de ușor folosit pentru crearea foilor de stil XSL.

PARTEA A IX-A

Tehnici avansate

Capitolul 40
Internaționalizarea limbajului HTML

de Dmitry Kirsanov

În acest capitol:

- **Standarde de codificare a caracterelor 624**
- **Standardul MIME 628**
- **Setul de caracter HTML 629**
- **Identificarea limbii 634**
- **Marcarea prezentării specifice unei limbi 636**
- **Fonturile 639**

Nici o carte despre limbajul HTML nu este completă fără un capitol care să treacă dincolo de pronunțata tendință vestică în limba engleză și să fructifice aplicarea acestuia într-un context mondial, multilingvistic. Acest capitol prezintă principalele moduri de abordare ale acestei problemei, atât cele folosite în practică de către webmasterii din toată lumea, cât și cele sugerate de către organizațiile de standardizare.

Problema principală a internaționalizării limbajului HTML (sau *i18n*, cum este prescurtat uzual: litera *i*, urmată de numărul 18 și litera *n*) este redarea corectă a caracterelor folosite de alte limbi. De aceea încep prin examinarea diferitelor standarde pentru codificarea caracterelor (seturi de caractere). Aceste standarde sunt clasificate după lungimea combinațiilor de biți pe care le folosesc, începând cu codul ASCII pe 7 biți și până la Unicode și ISO 10646.

Problemele internaționalizării limbajului HTML au fost sintetizate pentru prima dată în importantul document RFC 2070 (`http://ds.internic.net/rfc/rfc2070.txt`). Prevederile din RFC 2070 au fost apoi încorporate în versiunea 4.0 de HTML. Cea mai mare parte a dezbaterilor din acest capitol se bazează pe proiectul de specificație pentru HTML 4.0 (vezi `http://www.w3.org/TR/WD-html40/`).

Capitolul continuă cu investigarea noului set de caractere pentru documente, definit pentru HTML 4.0. Vi se prezintă deosebirea importantă dintre setul de caractere pentru documente și codificarea externă a caracterelor. Veți afla de asemenea și despre metodele existente pentru specificarea codificării externe a caracterelor, adăugirile propuse pentru introducerea datelor în mai multe limbi, ca și unele probleme reale, referitoare la setul de caractere HTML.

Un alt aspect esențial al internaționalizării limbajului HTML este marcarea limbii – adică specificarea limbii în care e scris un text pentru a ajuta programul agent pentru utilizator să-l redea simultan cu observarea convențiilor tipografice ale limbii respective. Sunt tratate în versiunea 4.0 a limbajului HTML și unele aspecte privind limba în care sunt prezentate textele, prin instrumente pentru controlul direcției de scriere, legarea caracterelor cursive, redarea ghilimelelor, alinierea textului și despărțirea în silabe. În concluzie, voi prezenta pe scurt problemele legate de folosirea fonturilor în internaționalizarea limbajului HTML.

Standarde de codificare a caracterelor

Faptul că industria calculatoarelor a înflorit în țara a cărei limbă folosește unul dintre cele mai compacte alfabete din lume a fost o întâmplare. Totuși, la puțin timp după ce primele calculatoare au învățat să citească pe litere în limba engleză, a fost nevoie și de codificarea caracterelor din alte limbi. De fapt, chiar setul minim de litere latine și simboluri de bază au fost subiectul controversei dintre două standarde aflate în competiție, ASCII și EBCDIC (acum aproape dispărut). Nu este de mirare că pentru alfabetele altor limbi a existat o confuzie similară pentru o perioadă și mai mare de timp – și situația e încă departe de a fi lămurită.

Fiecare codificare a caracterelor (numită frecvent *set de caractere* sau, mai exact, *setul de codificare al caracterelor*) e definită în primul rând de intervalul numeric al codurilor; în al doilea rând de colecția de caractere; și în al treilea rând de corespondența dintre acestea. De aceea, termenul set de caractere e puțin confuz, întrucât acesta presupune două seturi precum și relația dintre acestea. Definiția cea mai exactă a codificării caracterelor e dată probabil de Dan Connolly în lucrarea sa „Character Set Considered Harmful" (Setul de caractere considerat periculos") (`http://www.w3.org/pub/WWW/MarkUp/html-spec/charset-harmful.html`): „Funcția al cărei domeniu de definiție este un subset de numere întregi și al cărei domeniu de valori este un set de caractere".

Domeniul codurilor este limitat de lungimea șirului de biți (numit *combinația biților*), folosit pentru codificarea unui caracter. De exemplu, o combinație de 8 biți este suficientă pentru a codifica în total 256 de caractere (deși este posibil să rămână poziții nefolosite ale codului). Cu cât este mai mică dimensiunea combinației de biți, cu atât este mai compactă codificarea (cu atât este mai mic spațiul necesar pentru stocarea unui text), dar cu atât este mai mic numărul total de caractere pe care le puteți codifica.

Este logic ca să se codifice caracterele folosind combinații de biți cu o dimensiune convenabilă pentru calculatoare. Datorită faptului că arhitectura modernă a calculatoarelor e bazată pe octeți, toate standardele de codificare actuale folosesc combinații de biți cu lungimea de 8, 16 sau 32 de biți. Următoarele subcapitole descriu cele mai importante astfel de standarde pentru a înțelege rolul jucat de acestea în Internetul zilelor noastre.

Codul ASCII pe 7 biți

Codificarea ASCII pe 7 biți, sau US ASCII, este echivalentă cu standardul internațional denumit ISO 646 (`ftp://dkuug.dk/i18n/ISO_646`), stabilit de Organizația Internațională pentru Standardizare (ISO). Această codificare folosește de fapt 8 biți pentru un caracter, dar primul bit, cel mai semnificativ, al fiecărui octet rămâne nefolosit, fiind întotdeauna zero. Cei 7 biți utili ai standardului ISO 646 pot codifica un total de 128 de caractere.

ISO 646 este standardul de codificare folosit de marea majoritate a calculatoarelor din lumea întreagă (fie codificare propriu-zisă, fie ca parte a altor codificări, după cum veți vedea în curând). ISO 646 poate fi denumit „internațional" în sensul că există foarte puține calculatoare în lume care folosesc alte codificări pentru aceeași colecție de bază de caractere. El mai este folosit în mod exclusiv pentru cuvinte cheie și pentru sintaxă în toate limbajele de programare și marcare (inclusiv SGML și HTML), ca și pentru tot felul de date ce pot fi editate de către oameni, dar care sunt în esență generate de calculatoare, cum ar fi de exemplu fișierele de configurare sau fișierele script.

Totuși, în ceea ce privește numărul mare de limbi naturale vorbite în lumea întreagă, ISO 646 este foarte restrictiv. De fapt, numai limbile engleză, latină și swahili pot folosi codul ASCII simplu, pe 7 biți, fără caractere suplimentare. Cele mai multe limbi ale căror alfabete (numite *scripturi* sau *sisteme de scriere*) au la bază alfabetul latin folosesc litere cu accent și ligaturi.

Primele 32 de coduri din ISO 646 sunt rezervate pentru *caracterele de control*, ceea ce înseamnă că acestea invocă unele funcții sau caracteristici ale dispozitivului care mai degrabă citește textul decât să producă forme vizibile (numite frecvent *glife*) de caractere pentru cititorii umani. Standardele pentru seturi de caractere au de regulă reținere în definirea precisă a funcțiilor caracterelor de control deoarece aceste funcții ar putea diferi considerabil, în funcție de natura programelor pentru procesarea textului.

De exemplu, din cele 32 de caractere de control din ISO 646, numai câteva (carriage return, line feed, tabulation) au semnificații mai mult sau mai puțin stabilite. Cea mai mare parte din aceste coduri este pur și simplu inutilă pentru a fi folosită în cadrul textelor. Acest spațiu de cod irosit este o reminiscență din timpurile în care aceste caractere de control jucau rolul actualelor formate ale documentelor și al protocoalelor de comunicații.

Codificarea pe 8 biți

Primul pas natural pentru adaptarea la limbi care au nevoie de mai multe litere decât limba engleză este să se folosească cel de-al optulea bit din fiecare octet. Astfel, obțineți încă 128 de coduri care sunt suficiente pentru a codifica un alfabet cu mai multe zeci de litere (de exemplu, chirilic sau grec) sau un set suplimentar de caractere latine cu semne diacritice și ligaturi folosite în multe limbi europene (cum ar fi ç în franceză sau β în germană).

Din nefericire, în lume sunt mult mai multe codificări pe 8 biți decât sunt necesare. Aproape fiecare platformă de calculator sau sistem de operare care a pătruns pe o piață națională fără o industrie proprie de calculatoare puternică a introdus un nou standard de codificare. De exemplu, acum în Rusia sunt folosite pe scară largă nu mai puțin de trei codificări pentru alfabetul chirilic, una fiind rămasă din zilele sistemului MS-DOS, cea de-a doua nativă în Microsoft Windows și cea de-a treia fiind populară în comunitatea UNIX sau în Internet. O situație similară există în multe alte comunități naționale ale utilizatorilor.

Fiind o instituție cu autoritate internațională, ISO a făcut tot ce i-a stat în putință pentru standardizarea dezordinii din codificarea pe 8 biți. Seria de standarde ISO 8859 (vezi `http://wwwwbs.cs.tu-berlin.de/~czyborra/charsets/`) conține aproape toate extensiile alfabetului latin ca și cel chirilic (ISO 8859-5), arab (ISO 8859-6), grec (ISO 8859-7) și ebraic (ISO 8859-8). Toate aceste codificări sunt compatibile cu standardul anterior, ISO 646; acest lucru înseamnă că primele 128 de caractere din fiecare tabel de cod ISO 8859 sunt identice cu cele din ASCII pe 7 biți, iar caracterele naționale sunt întotdeauna situate în cele 128 de poziții superioare ale codului.

Din nou, primele 32 de poziții ale codului (de la 128 până la 159 zecimal, inclusiv) din jumătatea superioară în ISO 8859 sunt rezervate pentru caracterele de control și nu ar trebui folosite în text. Totuși, de această dată, mulți producători de software au preferat să ignore interdicția; de exemplu, majoritatea fonturilor TrueType pentru Windows respectă ISO 8859-1 în ceea ce privește pozițiile de la 160 încolo ale codului, dar folosește domeniul 128-159 pentru caractere suplimentare (în special linia em și semnul de Trademark). Acest

lucru conduce la o confuzie interminabilă asupra faptului dacă aceste 32 de caractere pot fi accesate în HTML sau nu (definiția DTD, potrivit ISO 8859, declară această gamă de caractere nefolosite). Extensiile pentru internaționalizarea limbajului HTML rezolvă această controversă prin facilitarea adresării acestor caractere prin codurile lor Unicode.

Autoritatea ISO nu a fost totuși suficientă pentru a face din întreaga serie 8859 o alternativă puternică la codificările naționale de circumstanță oferite de sisteme de operare și platforme cunoscute. De exemplu, ISO 8859-5 e foarte rar folosit pentru codificarea textelor rusești, cu excepția unei mici părți din stațiile de lucru UNIX.

Pe de altă parte, primul standard din seria 8859, ISO 8859-1 (numit uzual ISO Latin-1), care conține cele mai răspândite extensii ale alfabetului latin, care deservesc multe limbi europene, a fost recunoscut pe scară largă ca fiind extensia ASCII pe 8 biți. Ori de câte ori este nevoie de un standard pe 8 biți care să fie cât de internațional este posibil, este probabil să întâlniți standardul ISO 8859-1 jucând acest rol. De exemplu, ISO-8859-1 a servit ca bază la elaborarea setului de caractere pentru documente până la versiunea 3.2 a limbajului HTML (la versiunea 4.0, rolul a fost preluat de Unicode, după cum vom vedea mai târziu, în subcapitolul „Setul de caractere HTML" din acest capitol).

Codificarea pe 16 biți

Nu toate limbile existente în lume folosesc litere mici în alfabet. Unele sisteme de scriere (de exemplu cel japonez și cel chinez) folosesc *ideograme* sau hieroglife în locul literelor, fiecare având corespondent nu un sunet din vorbire, ci un întreg concept sau cuvânt. Datorită faptului că există mult mai multe cuvinte și în special idei care pot fi concepute față de numărul sunetelor dintr-o limbă, astfel de sisteme de scriere conțin de obicei mai multe mii de ideograme. Codificarea unor astfel de sisteme necesită cel puțin 16 biți (2 octeți) pentru fiecare caracter, ceea ce vă permite realizarea unui total de 2^{16}=65536 caractere.

Ideal, această codificare pe 16 biți ar trebui să fie compatibilă cu codificările anterioare pe 8 biți (și în mod special cu ASCII pe 7 biți), ceea ce înseamnă că un dispozitiv ASCII care citește un astfel de șir de date să poată să interpreteze corect cel puțin caracterele ASCII, dacă acestea există. Această interpretare e realizată folosind *comutarea codului* sau tehnicile *escape*: secvențe speciale de caractere de control sunt folosite pentru comutarea între modurile ASCII cu un octet per caracter și cu doi octeți per caracter (denumite și *pagini de coduri*). Codificările bazate pe acest principiu sunt acum folosite pe scară largă pentru limbile Extremului Orient.

Comutarea codurilor funcționează bine, dar o problemă interesantă este aceea că această tehnică face ca reprezentarea codului unui caracter să fie lipsită de claritate – este oare codul pe 2 octeți sau codul precedat de secvența de comutare? Este evident că simbolurile naționale „extinse" și caracterele ASCII nu sunt tratate la fel în astfel de sisteme, ceea ce ar putea fi argumentat în practică, dar poate pune probleme în viitor.

În ultima parte a anilor '80, necesitatea unui standard cu adevărat internațional de codificare pe 16 biți a devenit evidentă. Consorțiul Unicode (`http://www.unicode.org`), care a luat ființă în anul 1991, a creat un astfel de standard, numit *Unicode*. În Unicode, fiecărui caracter din cele mai importante sisteme de scriere din lume i se asociază un cod unic pe 2 octeți. Potrivit tradiției pe care am menționat-o mai devreme, primele 128 de coduri Unicode sunt identice cu cele ASCII pe 7 biți, iar primele 256 de coduri, cu ISO 8859-1. La drept vorbind, acest standard nu este totuși compatibil cu codificările anterioare pe 8 biți; de exemplu, corespondentul Unicode al literei latine A este 0041(hexazecimal), dar în codul ASCII este numai 41.

Standardul Unicode necesită o carte separată pentru a-l descrie în întregime (de fapt specificația sa oficială a fost editată de Unicode Consortium sub forma unei cărți). Numărul mare de blocuri și zone ale acestuia cuprinde toate alfabetele literale și silabice aflate în uz, alfabete ale multor limbi moarte, o mulțime de simboluri speciale și caractere combinate (cum ar fi litere cu toate semnele diacritice posibile, cifre încercuite ș.a.m.d.).

Un dezavantaj al acestei abundențe de caractere este problema comparării șirurilor de caractere. De exemplu, Unicode conține atât litera accentuată é (codul hexa 00E9), cât și părțile sale constituente: accentul ascuțit ' (cod hexa 00B4) și litera mică e (cod hexa 0065). De regulă, dacă punem semnul accentului înainte și litera neaccentuată după, ar trebui să obținem același rezultat vizual ca în cazul formei combinate a literei, așa că apare întrebarea dacă într-adevăr calculatorul trebuie să considere aceste reprezentări ca fiind egale. Soluția generală acceptată de HTML 4.0 este că șirurile de text sunt considerate egale numai dacă au în compunere exact aceleași caractere Unicode, în aceeași ordine, chiar dacă este neglijată diferența dintre literele mari și mici.

Unicode oferă spațiu pentru peste 20000 de ideograme unificate, folosite în limbile Extremului Orient. Spre deosebire de alte alfabete, sistemele ideografice au fost tratate independent de limbă. Acest lucru înseamă că o ideogramă care are semnificații și aspecte similare în diferite limbi orientale, e reprezentată prin același cod unic, chiar dacă ea corespunde de fapt unor *cuvinte* diferite din fiecare limbă și cea mai mare parte a acestor ideograme au variante de glife specifice fiecărei țări.

Sistemul ideografic rezultat, încorporat în Unicode, e adeseori prescurtat *CJK*, după numele principalelor limbi conținute de acest sistem (chineză, japoneză și coreană). Sistemul CJK de unificare reduce setul de ideograme care trebuie codificate la un număr care poate fi controlat (și codat), dar efectul secundar nedorit este acela că este imposibil să se creeze un singur font Unicode, care să poată fi folosit de toți; un text în chineză ar trebui să fie afișat folosind forme vizuale ale ideogramelor puțin diferite față de cele din textul japonez, chiar dacă folosesc aceleași ideograme codificate în Unicode.

Elaborarea standardului Unicode este departe de a fi completă; aproape 34% din spațiul total rezervat codului nu este atribuit. Grupuri de lucru din cadrul consorțiului Unicode și ISO își desfășoară activitatea pentru selecția și codificarea celor mai merituoși candidați pentru colonizarea terenurilor Unicode încă necultivate. Un semn bun este că a crescut simțitor acceptarea standardului Unicode în întreaga industrie a calculatoarelor; de exemplu, Unicode este folosit pentru codificarea internă a caracterelor în limbajul de programare Java și pentru macheta fonturilor în sistemele de operare Windows 95 și Windows NT.

Standardul ISO 10646

Deși Unicode nu este încă folosit pe scară largă, ISO a publicat în anul 1993 un nou standard de codificare, pe 32 de biți, numit ISO/IEC 10646-1 sau setul universal de caractere codificate pe mai mulți octeți (Universal Multiple-Octet Coded Character Set, abreviat cu *UCS*, vezi `http://www.dkuug.dk/JTC1/SC2/WG2/docs/standards`). Asemănător standardului ASCII pe 7 biți, acest standard lasă nefolosit bitul cel mai semnificativ din octetul cel mai semnificativ, ceea ce-l face să fie în esență o codificare pe 31 de biți.

Totuși, spațiul rezervat codurilor ISO 10646 se extinde până la suma impresionantă de 2^{31}=2147483648 de poziții, ceea ce este mult, mult mai mult decât ar putea fi folosit de către toate limbile și sistemele de scriere care au existat vreodată pe Pământ. Care este atunci motivul ascuns al existenței unui astfel de „Unicode de Unicoduri"?

Principalul motiv al dezvoltării unui standard de codificare pe 4 octeți este faptul că Unicode nu poate acoperi de fapt *toate* caracterele pentru care codificarea ar fi utilă. Deși o parte semnificativă a pozițiilor Unicode este încă nealocată, propunerile pentru grupuri noi de caractere și ideograme care sunt luate în considerare acum necesită un număr de câteva ori mai mare de poziții pentru cod față de câte sunt disponibile în standardul Unicode pe 16 biți.

Deci extinderea setului Unicode pare inevitabilă și ar fi lipsit de sens să fie extins cu un octet, deoarece calculatoarele vor avea probleme cu prelucrarea secvențelor de 3 octeți (24 biți); pe de altă parte, codificarea pe 32 biți este convenabilă în special pentru calculatoarele moderne, majoritatea acestora procesând informația în porțiuni de 32 de biți.

Așa cum Unicode reprezintă o extensie a standardului ISO 8859-1, așa și noul standard ISO 10646 este o extensie la Unicode. Potrivit standardului ISO 10646, o secvență de 256 de poziții de cod poartă numele de *rând*, 256 de rânduri constituie un *plan*, iar 256 de planuri reprezintă un *grup*. Întregul spațiu alocat codurilor este deci împărțit în 128 de grupuri. Potrivit acestor termeni, Unicode este de fapt planul 00 al grupului 00, planul special denumit în ISO 10646 *planul de bază multilingvistic (Basic Multilingual Plane*, BMP). De exemplu, litera A din alfabetul latin (Unicode 0041) este codificată complet prin 00000041 în ISO 10646. Până acum ISO 10646 BMP este absolut identic cu Unicode și este puțin probabil ca aceste două standarde să difere vreodată.

ISO 10646 specifică unele formate intermediare, care nu necesită folosirea codurilor în *forma canonică* de 4 octeți pe caracter. De exemplu, setul UCS-2 (setul de caractere universal - Universal Character Set, pe 2 octeți) este identic cu Unicode atâta timp cât se folosesc codurile de 16 biți din planul BMP. Formatul UTF-8 (formatul de transformare UCS – UCS Transformation Format, pe 8 biți) poate fi folosit pentru a include, cu ajutorul unui procedeu de comutare a codului, codul pe 32 de biți într-un șir care conține în cea mai mare parte coduri ASCII pe 7 biți. Și în sfârșit, metoda UTF-16 a fost dezvoltată pentru a accesa peste un milion de coduri pe patru octeți din șirul de date Unicode/BMP pe 2 octeți, fără a-l face incompatibil cu implementările curente de Unicode.

Este foarte probabil ca ISO 10646 să fie rar folosit în forma sa canonică pe 4 octeți. Pentru majoritatea aplicațiilor care folosesc și procesează texte, irosirea a 32 de biți pentru un caracter este mult peste nivelul acceptabil al redundanței. Totuși, ISO 10646 este un standard important, deoarece stabilește o singură autoritate în teritoriul vast care se întinde dincolo de Unicode, prevenind astfel incompatibilitatea codificărilor multi-octet chiar înainte ca această problemă să apară.

Standardul MIME

MIME, al cărui nume vine de la *Multipurpose Internet Mail Extensions* (extensii multifuncționale pentru poșta Internet), este un standard dezvoltat inițial pentru extinderea posibilităților poștei electronice astfel încât să permită mesajelor de poștă electronică să cuprindă aproape toate tipurile de date și nu numai text simplu. Totuși, mecanismele MIME s-au dovedit așa de utile și de bine proiectate încât ele sunt folosite și în multe alte domenii, inclusiv în limbajul HTML. Puteți găsi ultima specificație MIME în documentele RFC, de la 2045 la 2049. (Vezi `http://ds.internic.net/rfc/rfc2045.txt`, `http://ds.internic.net/rfc/rfc2046.txt` ș.a.m.d.).

Sistemele existente de transport pentru poșta electronică, cum ar fi Simple Mail Transfer Protocol (SMTP) și Post Office Protocol 3 (POP3), nu acceptă decât text simplu în corpul mesajului. Acest lucru înseamnă că mesajul ar trebui să conțină numai caractere tipăribile

(nu caractere de control) din codul ASCII pe 7 biți, iar liniile trebuie să aibă o lungime rezonabilă. Pentru a depăși această limitare, standardul MIME introduce unele metode pentru conversia datelor binare sau a textului codificate pe mai mult de 7 biți în text simplu ASCII „protejat poștal".

Pentru a specifica ce metodă de conversie a fost folosită (dacă s-a folosit) și care a fost tipul original al datelor expediate în mesaj, la mesaje sunt adăugate câmpuri antet speciale MIME. Pentru mesajele text, este specificată codificarea caracterelor (setul de caractere sau *charset*) din corpul mesajului împreună cu alți parametri. Acest mecanism este important, deoarece este folosit nu numai pentru mesajele din poșta electronică, dar și în anteturile HTTP pentru documentele HTML transferate prin rețea. Parametrul `charset` este o parte a câmpului `Content-Type` și are următoarea formă:

```
Content-Type: text/html; charset=ISO-8859-1
```

În acest caz, `text/html` este identificatorul standard al tipului datelor din „sursa HTML", iar `ISO-8859-1` indică ce tip de codificare a caracterelor folosește textul din documentul HTML. Ambele valori sunt luate din registrul oficial al tipurilor de date conținute, al seturilor de caractere și al altor clasificatori legați de standardul MIME, care sunt întreținuți de IANA (autoritatea pentru alocarea numerelor în Internet – Internet Assigned Number Authority).

Acest registru oficial este ceea ce face standardul MIME atât de util în afara domeniului poștei electronice. Este important în special faptul că MIME a dezvoltat, în interesul dumneavoastră, un mod standard de comunicare a codificării caracterelor unui document. Puteți obține lista MIME a valorilor înregistrate `charset` de la adresa `ftp://ftp.isi.edu/in-notes/iana/assignments/character-sets`.

Setul de caractere HTML

După ce v-ați familiarizat cu standardele de codificare a caracterelor și metoda MIME pentru indicarea standardului folosit, este timpul să ajungem la limbajul HTML și să vedem cum a fost modificat în versiunea 4.0 pentru a lucra cu date din diferite limbi.

Comparație între setul de caractere de document și codificarea externă a caracterelor

Trebuie făcută mai întâi o distincție importantă. După cum cunoașteți din Capitolul 4, „Definirea tipului de document HTML", sintaxa HTML este definită de o definiție a tipului de document (Document Type Definition – DTD) . Definiția DTD este însoțită de o altă construcție formală numită *declarația SGML*, a cărei secțiune `CHARSET` definește singurul *set de caractere de document* care va fi folosit de către toate documentele HTML care se conformează acestuia.

Pe de o parte, această convenție face ca alegerea setului de caractere de document să fie evidentă: trebuie să fie internațional, ceea ce înseamnă Unicode sau chiar mai bine, ISO 10646. Iată cum definește declarația SGML pentru HTML 4.0 setul de caractere de document (vezi Capitolul 4 pentru explicații asupra sintaxei):

```
CHARSET
     BASESET  "ISO 646:1983//CHARSET
               International Reference Version
               (IRV)//ESC 2/5 4/0"
     DESCSET  0   9   UNUSED
              9   2   9
```

```
          11   2    UNUSED
          13   1    13
          14   18   UNUSED
          32   95   32
          127  1    UNUSED
BASESET "ISO Registration Number 176//CHARSET
          ISO/IEC 10646-1:1993 UCS-2 with
          implementation level 3//ESC 2/5 2/15 4/5"
DESCSET 128  32     UNUSED
          160  65375  160
```

ISO 10646 e folosit aici în unul din formatele sale de transformare, și anume UCS-2, care este formatul pe 2 octeți identic cu Unicode.

Documentul RFC 2070 abordează mai amănunțit problemele și construiește setul de caractere al documentului numai pe baza formei canonice pe 4 octeți a standardului ISO 10646, fără nici o referire la ISO 646 (care este oricum un subset din ISO 106), cu o limită superioară a spațiului rezervat codului ridicată până la 2147483646.

Pe de altă parte, nu ar fi normal să ne așteptăm – și nici să cerem – ca toți autorii de HTML și producătorii de browsere să treacă la Unicode în următoarele luni (sau ani). Se pune întrebarea cum să beneficiem de avantajele folosirii Standardului Unicode fără să îi facem pe toți ceilalți să treacă la folosirea lui?

Această dilemă este rezolvată prin diferențierea setului de caractere al documentului și de *codificarea externă a caracterelor* documentului. Codificarea externă este aplicată documentului când este păstrat pe un server și transferat prin rețea; această codificare poate fi opțională, dacă este suficient să se codifice colecția de caractere a documentului, iar atât serverul cât și agentul software pentru utilizator pot să-l folosească adecvat.

După primirea documentului, agentul software pentru utilizator ar trebui să-l convertească din codificare externă în setul de caractere al documentului, așa că ulterior se face procesarea SGML și analiza marcajelor numai în acest set de caractere. Înainte de afișarea documentului analizat, agentul pentru utilizator îl poate recoda – de exemplu, pentru a fi compatibil cu codificarea cerută de sistemul de operare (pentru a apela serviciile de afișare) sau să corespundă codificării fonturilor care vor fi folosite pentru afișare.

Conversia documentului din codificarea externă în setul de caractere al documentului specificat în limbajul SGML este făcută din două motive evidente. În primul rând este necesar să ne asigurăm că toate caracterele care au o semnificație specială în limbajul HTML, cum ar fi literele din numele elementelor și caracterele < și >, sunt recunoscute în mod corect (deși este puțin probabil, codificarea externă poate remapa unele dintre aceste caractere la alte combinații de biți).

În al doilea rând, amintiți-vă că utilizatorii pot invoca caracterele prin folosirea referințelor acestora, cum ar fi `©` sau `©` pentru simbolul de copyright. Pentru ca aceste referințe să fie clare și să nu necesite schimbări în momentul în care documentul e recodat de la un set de caractere la altul, specificația precizează faptul că referințele trebuie întotdeauna să facă referire la setul de caractere al documentului – adică Unicode.

De exemplu, pentru a accesa litera chirilică majusculă EF printr-o referință caracter, trebuie să folosiți codul ei Unicode, caruia îi corespunde `Ф`, indiferent de codificarea caracterelor cu care lucrați pentru crearea documentului dumneavoastră. Nu are importanță dacă folosiți codificarea chirilică pe 8 biți KOI8-R, în care această literă are codul zecimal 230 sau ISO 8859-1, care nu conține deloc alfabetul chirilic. Un analizor (parser) HTML ar trebui să

rezolve întotdeauna referințele caracterelor conform setului de caractere al documentului (Unicode), doar pentru faptul că, în momentul procesării specifice HTML, documentul ar trebui să fie deja convertit în setul de caractere al documentului.

Iată câteva din avantajele folosirii standardului Unicode drept set de caractere de document, fiind separat de codificarea externă a documentului:

- Această soluție este compatibilă cu versiunile anterioare ale standardului HTML. Într-adevăr, referințele caracterelor din HTML 3.2, de exemplu, făceau referire la ISO 8859-1, care este un subset propriu al standardului Unicode. (În acest caz, nu prezintă importanță dacă se folosește un cod pe 8 sau pe 16 biți, deoarece zerourile de completare pot fi neglijate în valorile codurilor din referințele caracterelor).
- Soluția este și suficient de flexibilă. Cei care scriu într-un HTML internațional pot continua să folosească seturi de caractere care sunt compatibile software pe scară largă și acest lucru reduce efortul pentru limbile lor. În același timp, ei pot accesa în mod direct întregul spațiu Unicode rezervat caracterelor prin referințele caracterelor.
- În fine, implementarea procedeului nu ar trebui să dea bătăi de cap producătorilor de browsere. Standardul nici nu solicită ca agenții pentru utilizator (browserele) să poată afișa orice caracter Unicode, dar oferă în schimb unele soluții pentru cazurile în care browserul nu poate genera o glifă dintr-un anume cod Unicode (de exemplu, afișarea codului hexazecimal sau a unei pictograme speciale în locul caracterului).

Specificarea codificării externe de caractere

Pentru ca browserul să transforme corect documentul recepționat dintr-o codificare externă în setul de caractere al documentului, trebuie să cunoască mai întâi codificarea externă. Până acum, MIME este singurul mecanism standard capabil să comunice această informație. După cum s-a prezentat mai devreme în acest capitol, parametrul `charset` este inclus în câmpul `Content-Type`, care trebuie să fie o componentă a oricărui antet HTTP – adică trebuie să fie înaintea oricărui document trimis de protocolul HTTP. Acest câmp ar trebui inclus și în antetul unui mesaj de poștă electronică ce conține un document HTML. În prezent, nu există nici o cale care să indice codificarea caracterelor pentru un document HTML regăsit prin FTP sau dintr-un sistem de fișiere local sau distribuit.

Browserele obișnuite, cum ar Netscape sau Internet Explorer, recunosc parametrul `charset` și încearcă să treacă la setul de caractere cerut înainte de a afișa un document (în Netscape Communicator 4, de exemplu, puteți deschide un submeniu View | Encoding pentru a vedea lista seturilor de caractere acceptate). Dacă nu este specificată nici o valoare charset, se ia în considerare ISO 8859-1 și dacă acesta nu este setul de caractere pe care a dorit autorul să-l folosească pentru acest document, utilizatorul trebuie să ghicească pe ce codificare să comute manual, pentru a citi documentul. (Putem spune că necesitatea comutării manuale a setului de caractere se datorează mulțimii severelor Web care nu declară niciodată codificarea caracterelor din documentele pe care le furnizează).

Mai este totuși ceva de spus în legătură cu negocierea setului de caractere. Protocolul HTTP permite unui program client să listeze un număr de codificări ale caracterelor cu care poate lucra, în ordinea preferințelor, chiar într-o cerere HTTP, folosind câmpul `Accept-Charset`. Cu ajutorul acestui listing, serverul poate selecta versiunea corectă a documentului dintre cele disponibile sau îl poate transforma din mers în setul de caractere cerut. Standardul declară faptul că dacă nu e furnizată nici o valoare `Accept-Charset` în cerere, agentul pentru utilizator poate lucra cu oricare set de caractere. Din păcate, singurul browser care permite (în prezent) utilizatorului să precizeze valoarea `Accept-Charset` pe care să o introducă în cererile HTTP este Lynx (vezi `http://lynx.browser.org`).

O altă metodă pentru a indica o codificare externă a documentului este emularea câmpului Content-Type din antetul elementului META. De exemplu, pentru a specifica faptul că documentul dumneavoastră este codificat cu KOI8-R chirilic, ar trebui să introduceți următoarea etichetă în blocul HEAD al documentului HTML:

```
<META http-equiv="Content-Type" content="text/html; charset=KOI8-R">
```

Această opțiune este la îndemâna celor care nu pot sau nu doresc să schimbe configurarea serverului pe care este stocat documentul, dar are un dezavantaj evident: un astfel de document, dacă este convertit automat dintr-o codificare în alta, necesită schimbarea manuală a atributelor etichetei <META>. Indicația de codificare META este acceptată de majoritatea browserelor, dar atenție: contrar prevederilor standardului potrivit cărora valoarea charset din antetul HTTP, dacă există, ar trebui să anuleze emularea META, unele browsere dau prioritate valorii META furnizate.

Internaționalizarea formularelor

Când navigați prin Web, pe lângă faptul că descărcați informații, uneori mai și încărcați informații, folosind mecanismul formularelor HTML. În mod obișnuit, acest mecanism necesită corecții pentru a permite negocierea setului de caractere pentru datele trimise de agentul software pentru utilizator la server. Acest subcapitol prezintă noile caracteristici introduse pentru a îndeplini această cerință.

În limbajul HTML 4.0, se conferă etichetei FORM un atribut suplimentar ACCEPT-CHARSET, similar câmpului HTTP Accept-Charset, menționat în subcapitolul precedent. Principala diferență este aceea că atributul ACCEPT-CHARSET din HTML lucrează în mod diferit, precizând ce codificare a caracterelor poate *primi* serverul de la utilizator. Valoarea atributului ACCEPT-CHARSET este o listă de identificatori MIME pentru codificările caracterelor cu care poate lucra serverul, în ordinea preferințelor; în mod normal lista conține cel puțin codificarea externă a caracterelor pentru documentul respectiv.

Valoarea atributului ACCEPT-CHARSET poate fi folosită de browser în mai multe feluri:

- Browserul ar putea configura suprafețele pentru introducerea textului astfel încât textul care este introdus să fie afișat folosind glifele corespunzătoare. Acest nivel de asistență este minim (este implementat, de exemplu, în Netscape Navigator 3.0, deși acest browser folosește codificarea principală a documentului în acest scop, în locul valorii atributului ACCEPT-CHARSET) deoarece utilizatorul rămâne cu problema introducerii corespunzătoare a textului. Dacă sistemul de operare nu acceptă codificarea, ar putea fi necesar să folosească un driver specializat pentru tastatură sau să copieze și să lipească textul convertit anterior. În unele cazuri, autorul de HTML poate furniza chiar în document indicația privind codificarea acceptată într-un anumit câmp de intrare
- Mai mult chiar, browserul ar putea lua în considerare codificarea caracterelor acceptată de către sistemul de operare și să convertească, dacă e posibil (când codificarea acceptată de către sistem și codificarea acceptată de către server au colecții de caractere identice) și necesar (dacă cele două codificări nu sunt identice), textul introdus de către utilizator, înainte de a-l trimite. Această prevedere ar face-o inutilă pe precedenta, deoarece sistemul de operare are grijă de afișarea corespunzătoare a caracterelor în zonele de introducere a textului, în condițiile în care folosesc codificarea nativă a sistemului. Acest nivel de asistență este implementat în Microsoft Internet Explorer 4.0 și Netscape Communicator 4.0 (dar trebuie spus din nou, ambele browsere ignoră valoarea ACCEPT-CHARSET și consideră valoarea charset a formularului identică cu valoarea charset a documentului).

- Documentul RFC 2070 sugerează faptul că browserul poate restricționa gama caracterelor care pot fi introduse în zona text, în conformitate cu codificarea precizată. După părerea mea, acest lucru este cam fără rost, dacă nu este însoțit de una dintre celelalte două prevederi.

A doua parte a problemei internaționalizării formularelor este cum să se trimită datele din formular o dată cu informațiile pivitoare la codificare. Pentru prima din cele două metode de trimitere, POST, standardul MIME este încă o dată folositor. Puteți adăuga parametrul charset la câmpul antet Content-Type:application/x-www-form-urlencoded, care precede orice date trimise cu metoda POST, ca în exemplul următor:

```
Content-Type: application/x-www-form-urlencoded; charset=KOI8-R
```

Un astfel de câmp antet trebuie să fie folosit de către browser, dacă datele din formularul POST sunt trimise folosind codificarea KOI8-R.

Documentul RFC 2070 acordă întâietate altui procedeu, care folosește tipul conținutului multipart/form-data, propus în documentul RFC 1867 (vezi http://ds.internic.net/rfc/rfc1867.txt), pentru încărcarea fișierelor bazate pe formulare (prevederile RFC 1867 sunt și ele incluse în HTML 4.0). În cazul acestei metode, datele din formular nu sunt încapsulate sub forma unui URL și fiecare pereche nume/valoare poate avea atașat propriul parametru charset.În prezent, această metodă nu este acceptată de browserele obișnuite, iar HTML 4.0 recomandă folosirea tipului de conținut multipart/form-data numai dacă datele din formular conțin fișiere care să fie încărcate de către server.

Prin cealaltă metodă de transmitere a formularului, GET, datele sunt întotdeauna încapsulate chiar în adresa URL pe care browserul o trimite serverului. În principiu, adresele URL pot conține orice combinații de biți, cu condiția să fie codificați folosind notația %HH (vezi Capitolul 12, „Să înțelegem locatorii uniformi de resurse"). Totuși, potrivit unui citat din documentul RFC 2070 „textul trimis prin formular este compus din *caractere*, și nu din octeți" și nu este ușor să se introducă informațiile referitoare la codificarea datelor text într-o adresă URL (altfel decât furnizând un câmp suplimentar de intrare pe care utilizatorul îl va stabili manual, ceea ce ar fi destul de incomod).

În consecință, specificațiile HTML 4.0 precizează că metoda GET este „nerecomandată din cauza internaționalizării". Documentul RFC 2070 sugerează, în cazul metodei GET chiar, ca un software agent pentru utilizator să poată trimite datele în corpul cererii HTTP, în loc de URL, deși în prezent nici o aplicație nu acceptă acest procedeu. O altă soluție în ceea ce privește adresele URL ar putea fi folosirea unuia dintre formatele speciale din ISO 10646; în particular, formatul UTF-8 (vezi http://ds.internic.net/rfc/rfc2044.txt) păstrează toate caracterele ASCII pe 7 biți și codifică toate caracterele ne-ASCII, folosind numai octeții cu bitul cel mai semnificativ 1 - adică cei din afara gamei ASCII. Această caracteristică face ca UTF-8 să fie compatibil integral cu sintaxa adresei URL curente. Deoarece ISO 10646 este un superset al tuturor celorlalte codificări ale caracterelor, un șir într-un astfel de format nu necesită nici o specificație charset suplimentară (desigur, cu condiția ca serverul să aibă cunoștință de folosirea formatului UTF-8).

Probleme rezultate din practică privind seturile de caractere

Diferențierile dintre setul de caractere al documentului și codificarea externă nu reprezintă singura noutate în limbajul HTML. Orice referință a unui caracter numeric dintr-un document conform cu HTML 3.2 sau cu o versiune anterioară trebuie să facă referire la caracterele din setul Latin-1 (ISO 8859-1), indiferent de setul extern de caractere al documentului. Din păcate, această convenție este ignorată de multe browsere actuale, ceea ce conduce la consecințe nedorite (deși, să admitem, nu prea serioase în cazul HTML 3.2 fără extensii de internaționalizare).

De exemplu, codificarea caracterelor KOI8-R, după cum este definită în RFC 1489 (vezi `http://ds.internic.net/rfc/rfc1489.txt`), precizează codul 191 (zecimal) pentru caracterul ©. În standardul ISO 8859-1, același simbol are codul 169. În mod ideal, când o entitate mnemonică `©` sau o referință caracter `&#169` (care este rezultatul dezvoltării `©`, în conformitate cu definiția DTD din limbajul HTML) este folosită într-un document KOI8-R, browserul trebuie să-l convertească în concordanță cu setul de caractere ISO 8859-1 și să afișeze simbolul de copyright (de exemplu, prin accesarea poziției de cod 191 din fontul KOI8-R).

Totuși, datorită faptului că majoritatea browserelor nu mai țin minte nimic despre setul de caractere ISO 8859-1 după ce au fost comutate pe KOI8-R sau oricare codificare externă folosită pentru document, autorul de HTML nu se mai poate baza pe tabelul cu entitățile mnemonice de caracter Latin-1. Aceste entități sau referințe pentru caractere numerice lucrează garantat numai dacă documentul respectiv e creat (și vizualizat) în ISO 8859-1.

Ca un fel de soluție, creatorii mai multor fonturi chirilice KOI8-R pentru a fi folosite în Web au optat pentru mutarea simbolului de copyright de la codul standard prestabilit 191, la cel de inspirație Latin-1 169. După cum a spus Alan Flavell de la CERN (Consiliul European pentru Energie Nucleară), „Fragmentarea fontului în scopul de a ajuta un browser defect este o idee proastă". O dată cu recunoașterea pe scară largă a folosirii limbajului HTML internaționalizat, problemele ar putea să devină mai serioase, deoarece referințele caracterelor Unicode din documentele corespunzătoare pot ieși mult mai ușor din sincronizarea cu codificarea externă a documentului.

În principal, suportul pentru codificările nestandardizate ale documentelor în browserele cum ar fi Netscape Navigator 3.0 se reduce la posibilitatea de a comuta fonturile afișate, ca răspuns fie la parametrul `charset` din antetul HTTP, fie la selectarea unei comenzi de către utilizator – și cam atât. Ca rezultat, Netscape Navigator 3.0 nu poate afișa texte în limba rusă în codificarea KOI8-R cu fonturi chirilice KOI8-R instalate, chiar dacă lucrează într-o versiune rusească de Windows care furnizează fonturi chirilice pentru codificarea Windows.

Sunt mai multe probleme referitoare la codificarea caracterelor de document pe care multe browsere obișnuite nu le pot rezolva și de aceea autorii de HTML ar trebui să acorde atenție următoarelor aspecte:

- Chiar dacă textul unui document e corect afișat, titlul acestuia, când conține caractere specifice pentru codificare, poate apărea fragmentat pe bara titlului ferestrei (aparent deoarece fontul folosit în barele de titlu pentru fereastră este determinat de sistemul de operare, care s-ar putea să nu știe nimic despre codificarea documentului).
- Textele `ALT` plasate în locul imaginilor inline, ca și etichetele butoane din formulare s-ar putea să nu fie afișate corect dacă nu conțin caractere specifice pentru codificare (din nou, cauza o reprezintă faptul că multe browsere folosesc fonturi furnizate de sistem în aceste scopuri).
- Appleturile Java orientate pe text din browserele ce acceptă limbajul Java s-ar putea să aibă probleme la afișarea textelor cu o codificare nestandardizată.

Identificarea limbii

Problemele privind setul de caractere reprezintă numai o parte din subiectul internaționalizării limbajului HTML. Aproape la fel de importantă este problema identificării limbii documentului. Multe aspecte legate de prezentarea documentului depind nu numai de setul de caractere, dar și de limba în care este scris textul respectiv.

De exemplu, după cum am spus anterior, aceleași ideograme sunt folosite în multe limbi orientale, astfel încât în fiecare limbă sunt redate prin glife puțin diferite și sunete diferite în pronunție. Asemănător, limbi diferite care folosesc același set de caractere ar putea diferi foarte mult prin modul de despărțire în silabe, folosirea spațiului, punctuației ș.a.m.d.

Pentru aceasta, limbajul HTML 4.0 a introdus un atribut nou `lang`, care poate fi folosit împreună cu majoritatea elementelor HTML, pentru a descrie limba conținutului elementului. „Limba" e definită în acest context ca fiind „vorbită (sau scrisă) de ființe umane pentru comunicarea informațiilor către alte ființe umane; limbajele calculatoarelor sunt excluse în mod explicit". Iată un exemplu de folosire a atributului `lang`:

```
<P lang="fr">Ce paragraphe est en Français</P>
```

Atributul `lang` poate avea ca valoare un cod prescurtat de două litere (sau *etichetă*) corespunzător limbii. Lista acestor coduri este definită de standardul ISO 639 (vezi `ftp://dkuug.dk/i18n/ISO_639`); aceste coduri nu trebuie confundate cu codurile corespunzătoare țării (de exemplu, codul de limbă `uk` înseamnă ucraineana și nu United Kingdom – Marea Britanie).

Pot fi utilizați și *identificatorii extinși* pentru a desemna diferite dialecte sau sisteme de scriere ale limbii, pentru a identifica țara unde este folosită ș.a.m.d. Acești identificatori extinși au la bază un cod de două litere la care se adaugă subcoduri separate prin linioară (-), după cum se vede în exemplele următoare:

- `en-US`

Limba engleză folosită în S.U.A. (subcodurile de două litere sunt întotdeauna interpretate ca fiind coduri de țară, în conformitate cu standardul ISO 3166)

- `no-nynorsk`

Varianta Nynorsk a limbii norvegiene

- `az-cyrillic`

Limba azeră scrisă cu litere chirilice

Baza de date cu acești indicatori extinși ai limbajului e întreținută de către IANA (vezi `ftp://ftp.isi.edu/in-notes/iana/assignments/languages/`). Toate valorile `lang` nu țin cont de tipul caracterelor; sintaxa lor completă este definită în documentul RFC 1766 (vezi `http://ds.internic.net/rfc/rfc1766.txt`). O altă resursă utilă este documentul de la adresa `http://domen.uninett.no/-hta/ietf/lang-chars.txt`, unde se găsește o listă a celor mai cunoscute limbi, împreună cu setul de caractere pe care-l folosesc.

Specificațiile de limbă sunt moștenite de la elementele părinte. De exemplu, dacă adăugați un atribut `lang` la eticheta `<HTML>` a documentului dumneavoastră, toate elementele din document au asociată aceeași limbă, cu excepția cazului în care pentru unele dintre ele se specifică o limbă separată. De asemenea, dacă nu se face specificarea în mod explicit a limbii în document, agenții pentru utilizator consultă câmpul antet MIME `Content-Language`, dacă este furnizat pentru acest document de către serverul Web. De exemplu, puteți configura serverul dumneavoastră să trimită următorul câmp antet, împreună cu documentele dumneavoastră:

```
Content-Language: en-US
```

Valoarea prestabilită pentru limba conținutului elementului este `unknown` (necunoscută).

Marcarea prezentării specifice unei limbi

Într-un mediu multilingvistic, s-ar putea să aveți nevoie să precizați unele aspecte ale modului de prezentare a textului în HTML, cum ar fi direcția de scriere (de la stânga la dreapta sau de la dreapta la stânga), particularități ale punctuației ș.a.m.d. Aceste aspecte pot fi deduse din limba textului (vezi subcapitolul precedent), dar uneori s-ar putea să aveți nevoie să precizați aceste informații fără a specifica limba sau să anulați valorile predefinite pentru limbă. Alte aspecte privind prezentarea (cum ar fi glilimelele) necesită marcarea suplimentară, chiar dacă este specificată limba.

Documentul RFC 2070 a introdus, iar HTML 4.0 a adoptat un set întreg de elemente, atribute și entități HTML noi, pentru acest tip de marcare a prezentării. Aceste caracteristici noi sunt prezentate pe scurt în subcapitolele următoare.

Direcția de scriere

Deși majoritatea limbilor vestice au scrierea de la stânga la dreapta, unele limbi, precum araba sau ebraica, au scrierea de la dreapta la stânga. În situațiile în care acest text este amestecat cu texte cu scriere din direcția opusă (rezultând un text bidirecțional sau *BIDI*), este necesară o marcare specială pentru a rezolva incertitudinea.

Standardul Unicode are mai multe prevederi legate de direcția de scriere. Fiecare caracter Unicode are asociat parametrul *bidirectional category*, care poate lua diferite valori, cum ar fi de la stânga la dreapta (left-to-right), de la dreapta la stânga (right-to-left), separator numeric sau neutru (de exemplu spațiul alb). Unele caractere (cum ar fi parantezele) sunt marcate ca *oglindite*, în funcție de direcția textului (în cazul textului de la dreapta la stânga, paranteza deschisă are aspectul uneia închise din scrierea de la stânga la dreapta și viceversa).

Pentru a oferi suportul corespunzător, documentul RFC 2070 a introdus instrumente de marcare a direcției de trei tipuri. Primul tip este format din marcaje stânga-dreapta și dreapta-stânga, care se comportă identic cu spațiile cu lățime zero care au proprietăți de direcție corespunzătoare. Aceste marcaje sunt preluate direct din inventarul Unicode, astfel încât ele sunt implementate în HTML ca entități care se transformă în caracterele corespunzătoare Unicode:

```
<!ENTITY lrm CDATA "&#8206;" -- marcaj stânga-dreapta -->
<!ENTITY rlm CDATA "&#8207;" -- marcaj dreapta-stânga -->
```

Marcajele de direcție pot fi utilizate în cazul în care, de exemplu, ghilimelele duble (care nu au direcție proprie, dar nu sunt nici caractere oglindite) se află între un caracter latin și unul în ebraică; în această situație, locul real al ghilimelelor depinde de apartenența lor la șirul text stânga-dreapta sau dreapta-stânga. Prin plasarea unui marcaj de direcție invizibil (`‎` sau `‏`) într-una din părțile laterale ale ghilimelelor, puteți fi sigur că ghilimelele sunt flancate de caractere care merg în aceeași direcție, rezolvând astfel ambiguitatea.

Al doilea tip de marcare a direcției este reprezentat de noul atribut `dir` (a nu se confunda cu elementul `DIR`), care, asemănător atributului `lang`, poate fi folosit în aproape toate etichetele HTML, pentru a indica direcția de scriere a textului din conținutul elementului. S-ar putea ca uneori să fie nevoie să indicați direcția de bază a scrierii pentru un text; de asemenea, marcarea direcției în mod explicit este importantă când există două sau mai multe niveluri imbricate de text contra-direcțional (consultați specificațiile HTML 4.0 pentru exemplificare).

Cele două valori posibile ale atributului `dir` sunt șirurile `RTL` (de la dreapta la stânga) și `LRT` (de la stânga la dreapta). Precum în cazul proprietăților foilor de stil în cascadă (CSS), puteți utiliza atributul `dir` în cazul în care nici un element nu este discriminat în mod

normal prin folosirea elementului `SPAN`, ca un fel de container neutru. Dacă atributul `dir` este omis, elementele la nivelul blocurilor (dar nu și elementele inline) moștenesc direcția de scriere a elementelor părinte. Direcția prestabilită de scrire pentru întregul document HTML este de la stânga la dreapta. După cum este precizat în specificație, agenții pentru utilizator nu trebuie să folosească valoarea atributului `lang` pentru a determina direcția de scriere a conținutului elementului.

Pentru mai multă concizie, definițiile atributelor `dir` și `lang` sunt împachetate într-o singură entitate parametru în definiția DTD din limbajul HTML 4.0:

```
<!ENTITY % i18n
 "lang     NAME        #IMPLIED      -- valoarea RFC 1766 pentru limba –
  dir      (ltrirtl)   #IMPLIED      -- directia prestabilita a scrisului –"
  >
```

Ulterior, entitatea `%i18n` este adăugată declarațiilor `ATTLIST` pentru majoritatea elementelor HTML.

În final, cel de-al treilea mod de marcare a direcției este reprezentat de noul element `BDO`, la nivel de frază (`BDO` vine de la *anularea bidirecțională* - bidirectional override). E folosit când trebuie afișate într-o singură direcție caracterele dintr-un amestec de text de la stânga la dreapta și de la dreapta la stânga, anulând proprietățile intrinseci privind sensul de scriere al caracterelor. Pentru elementul `BDO`, `dir` este singurul atribut obligatoriu.

Legarea caracterelor cursive

În unele sisteme de scriere (în special cel arabic), glifa corespunzătoare unei litere poate fi diferită în funcție de context – dacă litera este precedată sau urmată de alte litere. Literele arabe sunt modelate după prototipul scrisului cursiv și când se trasează o literă din mijlocul unui cuvânt, aceasta este legată cu literele vecine, deci s-ar putea să arate diferit față de cazul în care este izolată.

De regulă, aceste diferențe sunt rezolvate de un software specializat pentru afișarea literelor arabe, dar uneori este nevoie să se controleze comportarea modului de scriere cursiv când doriți să arătați, de exemplu, o literă izolată cu legăturile cursive. În acest caz, codul Unicode oferă două caractere speciale, ambele invizibile și cu lățime zero; primul forțează legarea caracterelor alăturate în cazul în care nu ar trebui legate în mod normal, iar cel de-al doilea oprește legarea care ar trebui să aibă loc. HTML 4.0 furnizează mijloacele de accesare a acestor caractere în HTML cu entitățile mnemonice de caracter `‍` și `‌`:

```
<!ENTITY zwnj   CDATA     "&#8204;"     -- latime zero fara legatura -->
<!ENTITY zwj    CDATA     "&#8205;"     -- latime zero cu legatura -->
```

Ghilimelele

Există mai multe stiluri diferite pentru a reda ghilimelele care încadrează citatele scurte, din cadrul textului. Deși limba engleză folosește întotdeauna ghilimele " ", franceza utilizează «», iar limba germană preferă „". Mai mult chiar, comentariile imbricate folosesc uneori stiluri diferite; de exemplu, în limba rusă sunt tradiționale ghilimelele franțuzești (fără spații de separare), în partea de sus și ghilimele germane pentru citate în citate. În final, este de dorit să fie posibil să se redea același text cu ghilimele variate într-un mediu grafic, dar cu ghilimelele duble simple din codul ASCII pe 7 biți pentru cazul browserelor în modul text.

Pentru a ține cont de aceste diferențe, documentul RFC 2070 a propus un nou element `Q` la nivelul frazei, al cărui conținut este înconjurat de o pereche de ghilimele redate în concordanță cu limba textului, nivelul de imbricare și posibilitățile de afișare, după cum se vede în exemplul următor:

```
<P lang="en">The English language always uses quotes <Q> like this</Q>,
French has <Q lang="fr">comme &ccedil;a</Q>,
and German prefers <Q lang="de">wie hier</Q>.</P>
```

Din păcate, această soluție nu este compatibilă cu versiunile anterioare de HTML; majoritatea produselor software existente ignoră etichetele `Q`, fără să afișeze nici ghilimelele simple ASCII, ceea ce poate altera adeseori semnificația textului. De aceea, specificația HTML 4.0, adoptând elementul nou, nu cere agentului pentru utilizator să pună întotdeauna între ghilimele conținutul unui element `Q`, dar trece această funcționalitate în sarcina foilor de stil, declarând:

„Este recomandat ca foile de stil să furnizeze o cale de a introduce ghilimelele înainte și după citatul delimitat de `Q` sau `BLOCKQUOTE` într-un mod adecvat contextului limbii curente... și gradului de imbricare al citatelor.

Totuși... este recomandat ca agenții pentru utilizator să nu introducă ghilimele în stilul predefinit.

Mai mult, dacă autorii includ ghilimele într-un element `Q` sau `BLOCKQUOTE`, agenții pentru utilizator nu trebuie să introducă ghilimele suplimentare".

Alinierea și despărțirea în silabe

Tradițiile în ceea ce privește folosirea modurilor de indentare a textului în alte limbi pot diferi de cele din limba engleză. Această diferență a determinat introducerea prin documentul RFC 2070 a atributului opțional `align`, care poate fi folosit cu majoritatea elementelor la nivel de bloc (și anume `P`, `HR`, de la `H1` la `H6`, `OL`, `UL`, `DIV`, `MENU`, `LI`, `BLOCKQUOTE` și `ADDRESS`), cu valorile `left`, `right`, `center` și `justify`. Documentul RFC 2070 recomandă ca valoarea implicită a atributului `align` pentru textele cu scriere de la stânga la dreapta să fie `left`, iar pentru textele de la dreapta la stânga trebuie să fie `right`, deși în specificația HTML 4.0 numai valoarea `left` este prezentată drept valoare implicită.

Acest atribut a fost îmbunătățit semnificativ față de versiunea 3.2 de HTML, unde lista de elemente care acceptau atributul `align` era mai scurtă (numai `DIV`, de la `H1` la `H6`, `HR`, `TD` și `P`), iar valoarea `justify` nu era permisă. HTML 4.0 abordează problema pe jumătate: adoptă opțiunea `justify` din RFC 2070, dar păstrează lista elementelor care acceptă atributul `align` aproape neschimbată față de HTML 3.2 (apar în plus elementele tabelă, cum ar fi `COLGROUP`, introduse în versiunea 4.0). De asemenea , HTML 4.0 recomandă folosirea foilor de stil și nu a atributului `align`.

În ceea ce privește despărțirea cuvintelor, se presupune că agenții pentru utilizator aplică reguli ce depind de limbă pentru a despărți cuvintele, dacă acest lucru este necesar pentru o afișare corespunzătoare. Pentru cazuri complexe sau importante, specificația HTML 4.0 recomandă ca autorii să folosească entități mnemonice `` care invocă folosirea caracterului soft al liniei de despărțire în silabe din Unicode ca și toată familia ISO 8859 și alte seturi de caractere.

Acest caracter invizibil marchează punctul în care se poate face despărțirea cuvântului. Când cuvântul este într-adevăr despărțit, caracterul este vizualizat ca o linie de despărțire normală (-). Din păcate, browserele obișnuite nu implementează această comportare; din păcate, atât Netscape Navigator cât și Microsoft Internet Explorer (chiar în versiunile 4) afișează *întotdeauna* simbolul - în locul unei linii soft, împiedicându-vă să folosiți acest caracter.

Pentru controlul mai bun al despărțirii cuvintelor, a fost propus noul element `HYPH`, care poate lucra în condiții complexe în care despărțirea unui cuvânt determină modificarea scrierii acestuia (de exemplu, cuvântul german *backen* devine *bak-ken* când este despărțit). Totuși, elementul `HYPH` nu a fost inclus nici în RFC 2070 și nici în HTML 4.0.

Fonturile

Fonturile sunt situate la granița dintre aspectele prezentării vizuale a documentelor HTML și problemele de internaționalizare ale limbajului HTML. Este prea puțin util ca HTML să accepte standardul Unicode, dacă nu se poate afișa catalogul său de caractere (sau cel puțin acea parte a codului Unicode pe care o utilizează documentul).

Bineînțeles, cei mai mulți utilizatori interesați în conținutul Web care nu este în limba engleză au deja instalate fonturile corespunzătoare în sistemele lor. Adesea, aceste fonturi sunt furnizate o dată cu versiunile locale ale sistemelor de operare sau împreună cu alt software și folosesc codificări care sunt cunoscute într-o anumită limbă. Browserele obișnuite, cum ar fi Netscape Navigator, permit folosirea acestor fonturi pentru vizualizarea paginilor Web.

Totuși, este necesară o metodă pentru a asigura afișarea corespunzătoare a datelor din mai multe limbi pe orice sistem. O soluție ar putea fi crearea și distribuirea unui pachet de fonturi gratuit (sau ieftin) pentru mai multe limbi sau a unui singur font cu structura Unicode a caracterelor. Un font Unicode, cu numele Cyberbit, este disponibil gratuit la adresa `http://www.bitstream.com/cyberbit.htm`, de la firma Bitstream.

Dezavantajul mare al soluției cu un singur font este că mărimea fișierului unui anumit font Unicode este de mai mulți megaocteți (chiar și fără zona ideogramelor). Cea mai practică soluție pentru situl Web din zilele noastre este probabil *serverul de glife*, un server proxy care înlocuiește structura bitmap inline pentru toate caracterele ne-ASCII, de pe pagina pe care o vizualizați. Prin intermediul acestui server se poate citi rapid o pagină într-o limbă străină, fără dureri de cap în privința fonturilor. Printre serverele de glife disponibile în prezent se află `http://www.lfw.org/shodouka` (numai japoneză) și `http://baka.aubg.bg`.

Ultimele versiuni de browsere cunoscute oferă totuși o altă soluție pentru această problemă. Atât Netscape Comunicator 4 cât și Microsoft Internet Explorer 4 acceptă unele variante ale *încapsulării fonturilor*, procedeul permițându-vă să trimiteți, o dată cu pagina Web, fonturile necesare pentru vizualizarea acesteia. Scopul principal al acestei extensii este îmbunătățirea calității tipografice a paginilor Web, deși încapsularea fonturilor, după cum veți vedea mai târziu, poate afecta în mod serios problema internaționalizării.

Implementarea făcută de către Microsoft a încapsulării fonturilor (vezi `http://www.microsoft.com/typography/web/embedding/default.htm`) folosește formatul fontului TrueType, care este larg răspândit. Fonturile încapsulate pot fi comprimate în mod eficient, pot fi grupate (sunt incluse numai caracterele folosite în pagină) și de asemenea pot fi protejate împotriva distribuirii neautorizate. Internet Explorer se bazează totuși în mare măsură pe serviciile de afișare ale fonturilor proprii sistemului de operare, ceea ce afectează portabilitatea. Soluția Netscape, care poartă numele de *font dinamic*, este bazată pe tehnologia TrueDoc a firmei BitStream (vezi `http://www.bitstream.com/world/`); are portabilitate mai mare deoarece browserul însuși se ocupă de afișare.

Deci, ce reprezintă încapsularea fonturilor pentru internaționalizare? Posibilitatea de a asigura afișarea corespunzătoare a oricărui caracter dintr-un document, pe orice sistem capabil să lucreze cu fonturi outline este un lucru pozitiv important. Totuși, există câteva capcane periculoase de-a lungul acestui drum.

Prima, unii autori de HTML (dar și producători de browsere), deoarece se pot baza pe fonturi furnizate, ar putea pătrunde în voie în zona ce conține seturi de caractere. De fapt, codificarea caracterelor unui document trebuie să respecte doar codificarea fonturilor care însoțesc documentul, ceea ce face redundante aproape toate prevederile internaționalizării

HTML care au fost descrise în acest capitol – și, în consecință, le pune în mare pericol să fie neglijate prin lipsa unui suport software. Bineînțeles, fonturile pentru distribuția pe Web pot folosi codul Unicode, dar nu există nici o garanție că întotdeauna va fi așa.

În al doilea rând, furnizarea fonturilor pentru HTML pune accentul și pe prezentarea vizuală a documentului, aspect care a fost deja evidențiat o dată cu extensiile curente proprii limbajului HTML. Multe documente din Web sunt create în prezent fără nici o grijă pentru portabilitate sau compatibilitate cu SGML și este puțin probabil că încapsularea fonturilor din documentele HTML, după cum sunt implementate de companiile care au realizat browserele cele mai folosite, va îmbunătăți această situație.

Pe de altă parte, este evidentă nevoia urgentă pentru controlul tipografic exact în HTML. Având în vedere acest lucru, consorțiul W3C a propus recent propria alternativă pentru soluțiile de încapsulare a fonturilor. Potrivit ideologiei de separare a structurilor de prezentare, această soluție (vezi `http://www.w3.org/TR/WD-font`) este o extensie a sistemului CSS, care furnizează descrierea detaliată, corespondența inteligentă și descărcarea fonturilor independentă de format. Cu această soluție, agenții pentru utilizator au patru căi de selectare a fonturilor pentru prezentarea elementelor HTML:

- *Corespondența exactă* a fontului specificat în foaia de stil, cu unul dintre fonturile instalate în sistem.
- *Corespondența inteligentă* a unui anumit font cu un font similar din sistem, dar diferit, dacă nu se poate face corespondența exactă.
- *Descărcarea* fontului din rețea, dacă cele două opțiuni precedente nu sunt disponibile, iar adresa URL a fonturilor este precizată.
- Ca o ultimă variantă, unii agenți pentru utilizator pot face *sinteza fonturilor*, pentru a crea din mers fonturile necesare pe baza descrierii fonturilor din foile de stil.

Viitorul internaționalizării limbajului HTML este în prezent destul de obscur. Standardele parcurse în acest capitol de-abia au fost finalizate și implementările lor în software sunt puține. De asemenea, marile companii de software sunt renumite pentru slaba compatibilitate cu standardele oficiale și prin urmărirea în schimb a propriilor extensii.

Totuși, majoritatea siturilor Web naționale se dezvoltă acum mult mai rapid decât întreaga rețea Web, astfel încât produsele Internet fără un minim de suport internațional vor fi foarte rare în curând. Din acest punct de vedere, prevederile limbajului HTML 4.0 și standardele legate de acesta au, pe lângă proiectarea bine gândită, avantajul de a fi deschise, independente și stabile.

INTERNAȚIONALIZAREA: UN STUDIU DE CAZ NEPREMEDITAT

Nu numai documentele HTML sunt bântuite de problemele internaționalizării. Orice document realizat cu calculatorul, care conține mai mult decât text ASCII, când este copiat dintr-un calculator în altul, este predispus la erori și incompatibilități. Chiar acest capitol al cărții reprezintă o ilustrare excelentă; l-am pregătit ca fișier Microsoft Word și l-am trimis la editori. Deși capitolul a fost în limba engleză, cele câteva caractere străine pe care le-am folosit ca exemple au dat multă bătaie de cap editorilor. Iată două note pe care și le-au scris editorii unul altuia și mie, încercând să-și imagineze care a fost semnificația unora dintre cei mai exotici octeți din fișierul meu și descrierea destul de ciudată a aspectului dorit al caracterelor, făcută de mine – nu ai altceva mai bun la dispoziție când nu există un standard comun la care să te referi.

Subiect: Ligatura germană ß

Dmitry: Este beta destul de apropiat de litera germană? Nu îl văd pe lista cu caractere. Trebuie să aleg dintre caracterele ANSI. – cm

CM: Caracterul „beta" din ANSI este similar cu litera germană, nu cu beta din greacă. ANSI nu conține alfabetul grecesc, deci este suficient să vă asigurați că luați caracterele dintr-un font ANSI și nu dintr-un font special precum Symbol (care conține întregul alfabet grec).–dk

Subiect: Exemple de diferite ghilimele

RJ: Nu știu ce să fac cu ghilimelele dinaintea expresiei *wie hier*. Codul pentru semnul dinainte de *comme ça* este [171], iar cel de după este [167] – din aceeași listă trimisă prin fax pentru Capitolul 3.–cm

CM: Nu pot să verific lista despre care vorbești aici, dar cred că vrei să spui 187, nu 167 – de fapt, ghilimelele din stânga (deschise) sunt codificate cu 171, iar cele din dreapta (închise) sunt codificate cu 187 și în ANSI și în Latin-1. Cât despre ghilimelele germane, ele lipsesc într-adevăr din aceste seturi de caractere, dar pot fi emulate în felul următor: ghilimelele germane deschise sunt o pereche de virgule („) apropiate una de alta și de cuvântul pe care-l cuprind, iar ghilimelele germane închise sunt identice cu cele deschise din limba engleză (de tipul „66").–dk

Rezumat

În acest capitol, am examinat prevederile privind internaționalizarea HTML, introduse în documentul RFC 2070 și mai apoi adoptate în HTML 4.0. Această parte a standardului HTML vă va permite să creați și să oferiți un conținut Web în oricare limbaj, folosind oricare sistem de scriere, mărind șansele ca audiența dumneavoastră să primească mesajul dumneavoastră și nu un amestec de caractere indescifrabile. Am prezentat următoarele aspecte în special:

- Ce este un set de caractere codificate, ce seturi de caractere standard există și cum sunt ele folosite
- Ce este standardul MIME și cum ajută el la comunicarea informației privind setul de caractere al unui document
- De ce este important să se facă diferența dintre setul de caractere pentru documentele HTML și codificarea externă a caracterelor unui document HTML
- De ce capcane ale browserelor uzuale trebuie să țineți seama când lucrați cu documente HTML internaționale
- Ce instrumente există în HTML 4.0 pentru a specifica limba oricărui fragment de document și parametrii proprii de prezentare specifici limbii (direcția de scriere, scrierea cursivă, redarea ghilimelelor, alinierea textului, despărțirea cuvintelor)
- Cum ar putea afecta problema accesibilității fonturilor viitorul internaționalizării limbajului HTML.

Capitolul 41
Crearea paginilor Web cu o mare accesibilitate

de Dmitry Kirsanov

În acest capitol:

- **De ce avem nevoie 643**
- **Cele două principii 644**
- **Textul 645**
- **Machetarea 648**
- **Grafica 651**
- **Formulare și legături 656**
- **Alte modalități 657**
- **O observație finală 659**

Pentru cei care nu au trăit experiența nici unei dificultăți în abilitățile lor fizice sau senzoriale, este foarte greu să-și imagineze cum percep lumea și schimbă informații cei care au trecut prin asta. Să admitem că nu sunteți obișnuit să întâlniți la oricare pas pe care-l faceți aspecte privind accesibilitatea. Dar, pe măsură ce intrați pe tărâmul virtual din Web, o astfel de atitudine trebuie reconsiderată.

Este dificil să supraestimezi impactul general al sitului Web asupra societății umane. Acesta este nu numai un nou mijloc de informare în masă, dar și un instrument capabil să ne facă pe fiecare dintre noi un furnizor de informații, dar și un consumator de informații. Accesibilitatea acestei tehnologii novatoare reprezintă o problemă de etică, chiar de politică, într-o măsură mult mai mare decât cea a profitului practic sau comercial. Având în vedere cele 750 de milioane de persoane cu handicap din lumea întreagă, nu este deloc nesemnificativă comparația dintre problema accesibilității în Web și principiul egalității în drepturi, indiferent de rasă, culoare, sex sau vârstă.

Această percepție a condus la lansarea, în aprilie 1997, a inițiativei de accesibilitate în Web (WAI), un proiect al consorțiului World Wide Web (W3C), cu scopul de a îmbunătăți standardele Web din punct de vedere al accesibilității (vezi `http://www.w3.org/WAI/`). Următoarele cuvinte sunt din declarația oficială de sprijin, semnată de președintele S.U.A., Bill Clinton:

„Datorită creșterii explozive a utilizării rețelei World Wide Web pentru publicare, comerț electronic, instruire pentru toate vârstele și livrarea serviciilor guvernamentale, este vital ca rețeaua Web să fie accesibilă pentru toți. Inițiativa de accesibilitate în Web va pune la punct instrumentele, tehnologia și liniile directoare pentru a face posibilă afișarea informațiilor în așa fel încât să fie accesibile tuturor utilizatorilor."

Acest capitol vă va face cunoștință cu considerațiile de bază, soluțiile existente și perspectivele de dezvoltare în domeniul accesibilității în Web. Veți învăța cum pot fi făcute paginile Web mai prietenoase cu persoanele cu handicap, cu tehnologia de astăzi și cu

investiții suplimentare minime sau nule. Aproape orice aspect al publicării în Web trebuie să țină seama de acest lucru, așa că, în acest capitol veți vizita din nou o mare parte din materia acestei cărți, păstrând în minte scopurile privind accesibilitatea.

De ce avem nevoie

Situl Web este este în primul rând un mijloc de informare vizual, așa că problemele de vedere sunt cele mai dificile în ceea ce privește accesibilitatea. Persoanele cu vedere slabă (reziduală) și cei care nu percep culorile folosesc diferite procedee pentru îmbunătățirea vizibilității, cum ar fi creșterea contrastului de culoare și folosirea fonturilor mari. Cei care sunt orbi trebuie să recurgă la sintetizoarele vocale sau ieșirea Braille (tactilă). Nu este nevoie să mai spunem că nici una dintre aceste metode nu poate păstra altceva decât strict esențialul conținutului de text al paginilor Web.

Unele persoane cu handicap fizic au probleme în folosirea dispozitivelor convenționale de intrare, cum este de cele mai multe ori mouse-ul. Este nevoie de un control motor mai fin pentru indicare-și-clic cu un mouse decât pentru apăsarea tastelor de pe tastatură, deci furnizarea unei modalități de intrare și navigare folosind numai tastatura este absolut necesară pentru interfața Web. Aceasta este, în cea mai mare parte, o problemă care se adresează celor care dezvoltă browsere, deși autorii de HTML pot fi și ei de ajutor (în special prin codificarea corespunzătoare a elementelor din formulare; vezi „Formulare și legături", din acest capitol).

În cazul altor tipuri de handicap, problemele de accesibilitate sunt mai puțin dificile. Utilizatorii surzi nu pot recepționa informația audio, dacă aceasta nu este dublată prin mijloace vizuale. Afecțiunile cognitive pot uneori afecta folosirea mecanismelor de navigare în pagină, chiar dacă este accesibil conținutul. Cerințele utilizatorilor de Web cu acces limitat sau acces numai la text au de asemenea legătură directă cu problemele de accesibilitate.

Există și situații în care persoanele care posedă toate abilitățile senzoriale, fizice și mentale preferă modul ne-vizual de livrare a informațiilor – de exemplu, ar putea fi convenabil să ascultați știrile când conduceți mașina sau mergeți pe jos. În sfârșit, ar trebui să vă amintiți despre roboți și agenții inteligenți care, sub multe aspecte, pot fi asemănați persoanelor cu handicap (de exemplu, roboții de indexare nu pot de obicei să extragă toate informațiile utile din imagini și trebuie să se bazeze doar pe text; pentru mai multe informații privind aceste lucruri, vezi Capitolul 42, „Strategii pentru indexare și motoare de căutare").

Autorii paginilor Web nu trebuie să furnizeze versiunea audio a întregului text de pe paginile lor. De obicei, utilizatorii cu handicap posedă un software și un hardware necesar (numite de obicei *tehnologii asistente*) pentru a lucra cu programele de calculatoare și cu datele, inclusiv cu paginile Web. De exemplu, un software pentru accesul vocal la paginile Web poate fi împărțit în *browsere vocale*, care interpretează sursele HTML de pe paginile Web (de exemplu, produsul pwWebSpeak: vezi `http://www.prodworks.com`) și produsele mai generale, numite *cititoare de ecrane*, care chiar citesc textul afișat pe ecran sau într-o fereastră (de exemplu, IBM Screen Reader, vezi `http://www.austin.ibm.com/sns/snssrd.html`).

Ați putea spune astfel că cel mai realist scop al unui autor de pagină Web este să prezinte conținutul într-un mod care este accesibil cu ușurință tehnologiilor asistente – în speranța că acestea vor satisface mai performant nevoile persoanelor cu handicap. Admitem că abordarea acestui scop ar putea uneori presupune luarea unor decizii justificate doar de caracteristicile unui anumit dispozitiv de acces. Multe dintre recomandările de accesibilitate pot fi totuși deduse din două principii de bază.

Cele două principii

Primul principiu al accesibilității este prezentarea informației prin *modalități alternative* – adică, folosirea a cât de multe medii și tehnici diferite de intrare/ieșire este posibil. De exemplu, canalul audio poate lucra pentru orbi, iar canalul vizual va dubla această informație pentru surzi. Principiul este important în special pentru elementele care sunt critice pentru livrarea mesajului, pentru înțelegerea a ceea ce va urma sau pentru navigarea către alte zone ale sitului.

În practică, principiul modalităților alternative devine echivalent cu imperativul de a furniza alternative sub formă de text pentru informația care nu este text. Aceasta deoarece textul simplu este mediul de informare cel mai natural pentru calculator; este ușor să fie afișat cu orice tip de interfață de calculator, este ușor de indexat și procesat automat și relativ ușor de convertit în vorbire. Prin prezentarea sub formă de text a informației, aveți avantajul „accesibilității intrinseci" a acestui format și nu trebuie să mai fiți preocupat deloc de modalitățile alternative.

Totuși, o pagină nu trebuie să conțină numai text pentru a fi accesibilă. Principiul modalităților alternative și prevederile limbajului HTML vă permit să realizați pagini care sunt atât bogate din punct de vedere al prezentării *cât și* accesibile. Dacă aveți nevoie de un clip audio, însoțiți-l de o transcriere în text; dacă doriți o imagine (cum doriți probabil cel mai adesea), asociați-i textul `alt`, care să transmită cât de multe informații este posibil privind imaginea (pentru mai multe informații privind textul alt, vezi subcapitolul „Imagini", din cadrul acestui capitol); iar dacă simțiți nevoia să folosiți un applet Java sau un obiect Shockwave, verificați dacă acest lucru este necesar *în mod critic* pentru navigarea prin paginile dumneavoastră sau pentru citirea textului (iar dacă nu este cazul, realizați un mecanism de acces alternativ pentru aceleași informații).

Cu alte cuvinte, o pagină Web accesibilă poate fi *multi-modală* (adică poate folosi procedee de intrare/ieșire diferite), cu condiția ca toate componentele sale critice să fie *alt-modale* (nici o parte a conținutului să nu fie redusă la o singură modalitate). Arta de a crea pagini Web alt-modale concomitent cu păstrarea eficienței, bogăției și eleganței acestora într-o măsură suficientă pentru a fi competitive în Web poate părea dificilă, dar este de fapt numai o problemă de respectare a unor reguli simple și evidente (majoritatea dintre ele sunt prezentate pe scurt în acest capitol).

Al doilea principiu fundamental al accesibilității este folosirea *logică* și nu *reprezentativă* a marcării conținutului (s-ar putea să cunoașteți aceste concepte din Capitolul 4, „Definiția tipului de document HTML", care face referire la rădăcinile SGML ale limbajului HTML). Marcarea logică înseamnă că pentru fiecare element al paginii dumneavoastră, trebuie să folosiți etichete HTML pentru a descrie ce *este* și nu cum *arată* (sau cum sună, sau apare la imprimantă, sau orice altceva).

Importanța marcării logice merge dincolo de problema accesibilității pentru persoanele cu handicap; de fapt, este cea mai bună metodă de a asigura în *general* accesibilitatea, posibilitatea de procesare și în final, longevitatea informațiilor dumneavoastră. Când calculatorul (nu numai cititorul uman) este conștient de structura logică a datelor, șansele de a fi capabil să convertească datele într-un alt format sau alt mediu sunt îmbunătățite foarte mult. Iar accesibilitatea implică, în cea mai mare parte, exact acest lucru: ușurința transferării documentului în alt mediu sau mod de prezentare pentru a face posibilă comunicarea acestuia persoanelor cu handicap.

Un exemplu este controversa dintre etichetele `I` și `EM`. Prima, care este un instrument de marcare reprezentativă, cere ca browserul să comute pe fontul cursiv și este limitat de

aceea numai la modul de prezentare vizual. Un browser vocal nu are „fontul cursiv", așa că va ignora probabil întreg segmentul de marcare. Pe de altă parte, `EM` este o etichetă logică care semnifică *evidențierea* și, deși un browser vizual poate interpreta acest element prin comutarea la fontul cursiv, un browser vocal îl poate citi cu voce tare și cu ton diferit („evidențiere acustică"). Un avantaj suplimentar al etichetei `EM` este că aceasta face distincția dintre stilul cursiv pentru evidențiere față de alte utilizări ale stilului cursiv, cum ar fi citatele (care sunt marcate cu eticheta logică `CITE`).

În mod normal, autorii de pagini Web au nevoie adeseori de instrumente pentru a ajusta și modul de prezentare al documentelor proprii – de exemplu, pentru a controla tipul fontului, mărimea sau tonul vocal. Răspunsul HTML la această cerere constă în *foile de stil*, un mecanism extern capabil să precizeze o multitudine de parametri de prezentare într-o manieră flexibilă (vezi `http://www.w3.org/TR/REC-CSS1` și Capitolul 18, „Utilizarea foilor de stil în cascadă", din această carte). Specificația curentă a foilor de stil HTML acceptă numai parametrii de prezentare vizuală, dar există o extensie propusă pentru lucrul și cu parametrii audio (de vorbire) (vezi `http://www.w3.org/TR/WD-acss`).

Textul

După cum am menționat anterior, din punctul de vedere al calculatorului, textul simplu este cel mai natural mod de prezentare a informației. Mai mult chiar, HTML este un limbaj orientat în mod intrinsec pe folosirea textului (amintiți-vă faptul că în abrevierea sa *T* vine de la „text"). De aceea, prin folosirea textului simplu de câte ori este posibil, veți crește accesibilitatea conținutului chiar fără eforturi suplimentare. De asemenea, în mai multe situații, accesibilitatea textului simplu poate fi îmbunătățită prin respectarea câtorva reguli simple.

Partea rămasă din acest capitol are ca sursă de inspirație un document important, „Reguli unitare de accesibilitate ale sitului Web" (Unified Web Site Accesibility Guidelines) pregătit sub autoritatea organizațiilor membre ale inițiativei de accesibilitate Web (vezi `http://www.trace.wisc.edu/HTMLgide/htmlgide.html`). Vă recomand să consultați acest document care nu este finalizat încă, pentru problemele referitoare la cele mai recente acorduri și materiale referitoare la accesibilitate.

OBSERVAȚIE

Pentru a verifica dacă documentul dumneavoastră respectă cerințele de accesibilitate, folosiți verificatorul online de accesibilitate, de la adresa `http://www.cast.org/bobby/`. Acest serviciu clasifică toate elementele paginii dumneavoastră și face recomandări asupra modului de creștere a accesibilității acesteia.

Marcarea și formatarea textului

Primul lucru pe care trebuie să-l țineți minte când marcați textul este să rămâneți adeptul etichetelor logice care descriu mai degrabă ceea ce doriți decât cum doriți să arate. Acest accent pe etichetele logice conduce la favorizarea unei întregi serii de elemente, inclusiv `TT` (fontul monospațiu), `I` (cursiv), `B` (aldin), `BIG` (fonturi mari), `SMALL` (fonturi mici), `STRIKE` (tăiate cu o linie), `S` (același cu `STRIKE`) și `U` (subliniate). Citând specificația, „deși unele sunt recomandate, folosirea lor este descurajată în favoarea foilor de stil" (elementul `BLINK` nu este introdus nici în limbajul standard HTML 4.0).

Același lucru se aplică binecunoscutului element `FONT` care a fost pentru mult timp mărul discordiei dintre cei în favoarea și cei în defavoarea extensiilor HTML pentru Netscape.

Elementele FONT și BASEFONT nu sunt recomandate în HTML 4.0, iar toate atributele acestora pun, fără excepție, probleme de accesibilitate. Atributul face este bineînțeles limitat doar la prezentările vizuale și s-ar putea să nu fie de ajutor nu numai pentru utilizatorii cu browsere de text sau vocale, dar și pentru browserele grafice care nu au un anume font instalat pe sistemele lor.

Atributul size al etichetelor FONT și BASEFONT este chiar mai periculos, deoarece pentru utilizatorii fără experiență sau mai neatenți, mărimea literelor poate să treacă drept înlocuitor al marcării logice. Este destul de obișnuit să vezi propoziții care trebuiau să fie anteturi, etichetate cu nimic altceva decât elementul FONT cu valoare mare a atributului size.

Această formatare nu poate avea un aspect decent într-un browser grafic, dar pentru accesul doar la text sau vocal, o propoziție cu această etichetă nu diferă de corpul textului, ceea ce are ca rezultat redarea ei în aceeași manieră cu ceea ce este înainte sau după. Un astfel de „antet" nu va fi marcat de nici o pauză sau schimbare de ton în vorbire, nici prin linii goale sau strălucirea caracterelor în modul text. De asemenea, aplicațiile de indexare automată sau de analiză nu vor avea acces la structura documentului dumneavoastră – ele nu vor putea spune unde se termină un capitol și unde începe celălalt.

OBSERVAȚIE

Uneori, autorii sunt atrași de prezentarea vizuală și nu logică, fără a scrie nici o singură etichetă. Împrumutate din stilul poștei electronice, emoticonurile (cum ar fi :-)), ca și compozițiile realizate cu săgeți (de exemplu, -->) se bazează pe modul cum arată aceste semne de punctuație, nu pe semnificația lor și de aceea nu sunt portabile într-un mediu nevizual. Imaginați-vă un orb încercând să înțeleagă ceva din „două puncte-linioară-paranteză închisă" citite fără intonație, cu voce tare, de browserul vocal. Acest lucru nu va fi prea plăcut, așa că vă rog să evitați orice semn de punctuație sau caracter fantezist.

Există totuși un contra-argument la această recomandare: multe browsere (inclusiv Netscape Navigator și Microsoft Internet Explorer până la versiunea 3) nu permit accesarea entității caracterului Unicode — pentru a produce o linie em (pentru a afla motivele pentru care nu puteți accesa acest caracter folosind entitatea — vedeți Capitolul 40, „Internaționalizarea limbajului HTML"). Astfel, pentru cei care nu mai vor să descifreze rebusuri ca „...este posibil, chiar probabil să aveți...", singura cale de a produce ceva care să semene rezonabil cu linia de unire este să ne bazăm pe -- sau ---, cu speranța că utilizatorii browserelor vocale pot ghici semnificația.

Culorile

Elementele FONT și BASEFONT, care au fost discutate în subcapitolele precedente, au un atribut în plus, color, care indică o culoare dorită a textului. Acest atribut, împreună cu atributele bgcolor, text, link, vlink și alink ale elementului BODY, pun o altă serie de probleme privind accesibilitatea (proprietățile foilor de stil referitoare la culori sunt de asemenea expuse la probleme similare, deși acestea pot fi mai ușor eliminate sau înlocuite de către utilizator).

Chiar și utilizatorii cu vederea normală sunt uneori jenați de alegerea culorilor care fac unele pagini aproape de necitit. Dacă fundalul este o culoare prea strălucitoare și aprinsă sau contrastul cu textul este prea redus, citirea textului devine o tortură. Un singur lucru poate fi mai rău: citirea unui text cu fonturi mici având o imagine cu multe detalii mici și contrastante drept fundal.

Alegerea greșită a culorii afectează în mod evident mai mult pe cei cu vedere slabă și cei care nu percep culorile, decât pe cei cu vederea normală. De aceea, trebuie să fiți foarte atenți la selecția culorilor și, dacă este posibil, să folosiți valorile implicite care nu pun nici

un fel de probleme. Dacă v-ați decis asupra unei combinații de culori mai puțin obișnuite, trebuie nu numai să verificați cum se citesc elementele, dar să o și testați pe monitoarele monocrome, pentru a verifica dacă nu ați confundat contrastul nuanțelor (care nu ar putea fi distinse de către persoanele care nu disting culorile) cu contrastul real (care e reprezentat mai ales de contrastul în strălucire și nu în nuanță).

Un monitor monocrome (sau versiunea nesaturată a imaginii paginii) nu reprezintă modelul real al nedistingerii culorilor pentru că multe persoane cu acest handicap nu sunt incapabile să perceapă *nici o* culoare, ci numai anumite culori, de obicei verde sau roșu. Aceste teste sunt obligatorii, datorită faptului că există mulți utilizatori de dispozitive de acces Web monocrome, calculatoare portabile sau de mână (pentru mai multe amănunte privind folosirea culorilor în proiectarea Web, vedeți http://www.webreference.com/dlab/9704/).

Ultimul lucru, dar nu cel mai lipsit de importanță, când se stabilesc culori nestandard în eticheta <BODY>, nu uitați să precizați în mod explicit *toate* atributele de culoare, inclusiv fundalul, textul, legăturile și legăturile vizitate. Unele persoane stabilesc propriile culori pentru fundal și text în browser, așa încât orice specificație parțială poate determina un contrast necorespunzător dintre culoarea textului de pe pagina autorului și culoarea fundalului de pe pagina vizitatorului (și invers). Nu uitați niciodată să stabiliți culoarea corespunzătoare pentru fundal, dacă folosiți imagini în fundal, altfel textul dumneavoastră ar putea deveni de necitit pentru acei utilizatori care au anulat opțiunea de autoîncărcare a imaginilor din browserele lor.

Listele

Listele sunt adesea folosite în documentele HTML pentru prezentarea colecțiilor pe un nivel sau a elementelor imbricate. O utilizare obișnuită a listelor, și anume rândurile de legături pentru navigarea prin sit (panouri de navigare), este în prezent implementată sub formă grafică; de asemenea, listele de elemente text folosesc uneori tabele pentru machetare și imagini grafice pentru marcaje de listă. Totuși, chiar în cazul în care sunt marcate de etichetele logice <OL> și <UL>, listele pot prezenta unele probleme privind accesibilitatea.

În cazul în care se vizualizează o pagină dintr-o fereastră a browserului, s-ar putea să nu vă dați seama cât de mult se bazează percepția dumneavoastră pe bidimensionalitatea reprezentărilor. Chiar în cazul în care citiți un text simplu, liniar, privirea dumneavoastră nu urmărește numai liniile, dar adeseori sare înainte și înapoi (involuntar de obicei), în încercarea de a găsi unde se termină paragraful sau subcapitolul curent, ce urmează ș.a.m.d. Informațiile dobândite din aceste salturi ale privirii permit creierului să planifice (involuntar, din nou) pentru viitorul imediat cum să crească eficiența citirii prin saltul peste ceea ce pare neimportant și prin „previzualizarea" materialului marcat vizual chiar înainte de a ajunge la el în cursul citirii.

Încercați acum să vă imaginați modul în care percepe aceleași informații un utilizator cu handicap, care este complet lipsit de indicii vizuale. Asemenea utilizatori nu au posibilitatea de a sări înainte (neștiind deloc unde să sară) și nici să revină înapoi la ceea ce a fost deja citit; memoria este aproape singurul lor mijloc. Pentru utilizatorii care posedă vedere bună, această limitare ar putea fi comparată cu citirea unei pagini printr-o gaură mică într-o foaie de hârtie (așa încât numai două cuvinte pot fi văzute prin ea), care se deplasează uniform de-a lungul liniilor textului.

Toate acestea vă ajută să înțelegeți cele două cerințe stabilite de acum pentru liste în documentul cu „Reguli unitare de accesibilitate", menționat anterior: primul, toate elementele dintr-o listă trebuie să fie numerotate și cel de-al doilea, ar trebui să precizați inițial câte elemente sunt în listă. Următorul exemplu prezintă cum pot fi utilizate cele două cerințe:

```
Sa recapitulam acum cele de mai sus printr-o lista de trei elemente:
<OL>
  <LI> Un element
  <LI> Alt element; urmează o sublistă de două elemente:
  <UL>
    <LI> Primul element
    <LI> Al doilea element
  </UL>
  <LI> Ultimul element
</OL>
```

După cum vedeți, nu trebuie să folosiți întotdeauna elementele `OL` (liste ordonate), deoarece uneori este mai potrivit să „numărați" elementele verbal, prin scrierea cuvintelor precum „primul", „al doilea" ș.a.m.d., în text.

Același lucru se aplică la declararea numărului total de elemente – nu trebuie să fiți pedanți și să urmați întocmai formulele date aici. Urmați cursul natural al textului dumneavoastră și faceți-l puțin mai explicit numai când este vorba de o listă. De exemplu, „cele două principii ale accesibilității pot fi prezentate pe scurt astfel:", va funcționa suficient de bine înaintea unei liste cu două elemente. Aceste prevederi vor face listele dumneavoastră să fie mult mai prietenoase cu utilizatorii cu handicap.

Machetarea

Machetarea este un concept total străin pentru limbajul HTML (cel puțin așa a fost imaginat de către creatorii limbajului). Deși versiunile moderne ale limbajului HTML permit specificarea unor parametrii de formatare vizuală (tipul fonturilor, de exemplu), plasarea unui element pe pagină este definită doar de locul în care intră acest element în textul sursei HTML. Excepțiile (precum atributul `align` pentru imagini și paragrafe) sunt rare și, în HTML 4.0, nerecomandate în favoarea foilor de stil (una din extensiile mecanismului foilor de stil realizează într-adevăr poziționarea bidimensională a elementelor paginii; vezi `http://www.w3.org/TR/WD-positioning`).

Totuși, autorii de pagini Web au nevoie de un mijloc eficient pentru aranjarea materialului pe pagină, iar foile de stil s-ar putea să nu constituie o soluție perfectă, deoarece nu sunt încă unanim acceptate. Acest lucru conduce la „inventarea" și dezvoltarea pe scară într-adevăr largă a instrumentelor „ad-hoc" de machetare, marea majoritate fiind tabele și imagini invizibile pentru spațiere. Pentru mult timp, această problemă a fost intens dezbătută: este permis să se folosească elemente HTML în afara destinației principale a acestora, doar de dragul efectelor lor secundare privind poziționarea?

Consensul pare să fie acela că unele dintre aceste „trucuri" sunt relativ inofensive pentru accesibilitatea documentelor, atâta timp cât sunt implementate cu grijă. Un exemplu de utilizare apreciabilă de acest fel poate fi văzut chiar pe pagina gazdă a consorțiului World Wide Web (`http://www.w3.org`), a cărui machetare se bazează pe tabelul cu două coloane (vezi Figura 41.1)

Tabelele

Care sunt metodele bune și cele greșite de a folosi tabelele pentru machetare? Răspunsul la această întrebare depinde de tipul dispozitivului de acces pe care-l aveți în vedere. Pentru majoritatea agenților pentru utilizatori ne-grafici, incluzând browserele în modul text, browserele vocale și sistemele automate de indexare, un tabel HTML nu reprezintă o aranjare bidimensională, ci o secvență liniară de celule, care se prezintă în ordinea în care sunt conținute în fișierul sursă HTML.

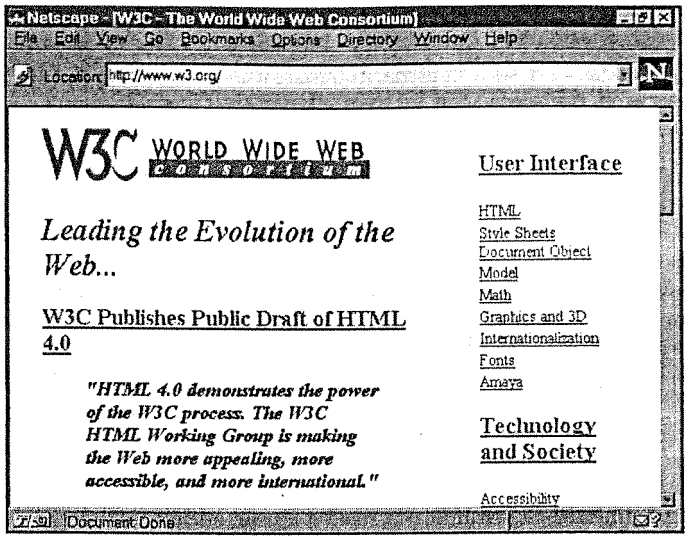


Figura 41.1. *Pagina gazdă a consorțiului W3C folosește tabelele pentru machetare.*

Pentru a avea întotdeauna un înțeles clar al conținutului, ar trebui să păstrați materialul orientat pe rânduri, cât de mult este posibil. Nu uitați că numai celulele care sunt una lângă alta *pe orizontală* vor fi citite în secvență; vecinele *pe verticală* vor fi probabil separate de alte celule.

De exemplu, nu puneți un titlu într-o celulă și textul care-i aparține în altă celulă de dedesubt, deoarece acest mod de așezare va avea probabil ca rezultat titlul separat de text, care nu are un înțeles clar de unul singur. Plasați în schimb și titlul și textul în aceeași celulă, folosind atributul `rowspan`, dacă este necesar, pentru a forța celula să se extindă în partea de jos cât este nevoie.

O cale rapidă de verificare a modului în care machetarea dumneavoastră respectă această cerință este să se scoată toată marcarea HTML și să se citească textul rămas în fișierul sursă căutând intervalele logice rămase goale și materialul rămas „agățat". Pagina gazdă de la W3C pe care o vedeți în Figura 41.1 este un exemplu de utilizare apreciabilă a tabelelor, deoarece toate părțile care sunt conectate în mod logic sunt în secvență în fișierul sursă (de exemplu, coloana din dreapta cu lista de legături este conținută în întregime într-o celulă).

Din păcate, titlurile separate de text nu reprezintă cel mai prost scenariu al dramei machetării-cu-tabele. Unele persoane care au nevoie ca o pagină Web să fie citită cu voce tare, folosesc în locul browserelor vocale specializate, browsere grafice convenționale combinate cu cititoare de ecrane – programe care pur și simplu citesc ceea ce este pe ecran (sau într-o fereastră). Un cititor de ecran nu are acces și nici nu înțelege sursa HTML de pe pagină, așa că poate doar să citească, de la stânga la dreapta și de sus în jos, tot conținutul unei ferestre a browserului, neglijând complet orice dispunere tabelară.

Documentul privind „Regulile unitare de accesibilitate" sugerează ca autorii de pagini Web să încerce să imite această comportare prin poziționarea unei bucăți de hârtie în fața monitorului și deplasarea lentă a acesteia în jos, citind ce apare pe întreaga lățime a ferestrei (o metodă mai simplă ar putea folosi fereastra unui alt program care este deplasată în jos). În mod normal, pagina gazdă de la W3C (vezi Figura 41.1) nu trece acest test; ceea ce veți auzi de la cititorul de ecran va fi un amestec de fragmente din coloane diferite: „W3C...User

Interface...HTML...Leading the evolution of the... Style Sheets...Document Object... Web... Model...Math...W3C Publishes Public Draft of HTML...Graphics and 3D" ș.a.m.d.

Nu știu dacă are sens să urmăriți întotdeauna accesibilitatea „strictă" în cazul cititoarelor de ecrane, deoarece acest lucru ar elimina total machetele multicoloană - ceea ce ar plăti într-adevăr un preț serios. Poate ar trebui să evităm folosirea coloanelor (ori să furnizăm în paralel o pagină numai cu text fără nici un artificiu de machetare) numai pentru acele pagini care sunt probabil accesate folosind cititoarele de ecrane.

Alte instrumente de machetare

Textul PREformatat cu fonturi monospațiu a fost larg folosit pentru prezentarea tabelară a materialului înainte de inventarea elementului TABLE. Este într-adevăr, până la o anumită limită, un element analog cu machetarea tabelară, deoarece elementul PRE permite controlul salturilor, indentărilor și alinierii în text. Trebuie să vă amintiți totuși că pentru utilizatorii cu handicap care au acces numai prin voce, orice poziționare și spațiu alb sunt inexistente; numai textul în sine poate fi perceput.

Practica de a folosi elementele BLOCKQUOTE pentru indentarea textului este de asemenea periculoasă (în HTML 4.0 nu este recomandată utilizarea acestora). Deși în cazul browserelor vizuale BLOCKQUOTE este cel mai adesea redat cu indentarea ambelor margini, ar trebui să vă amintiți că semnificația reală a acestei etichete logice este aceea de *citat*, și nu de indentare și există browsere care folosesc alte convenții de formatare (cum ar fi semnul > la începutul fiecărei linii). Întrucât un browser vocal nu are deloc mijloace de formatare, este foarte probabil ca să încadreze pur și simplu elementul BLOCKQUOTE în ceva de forma „început de citat...sfârșit de citat" – ceea ce nu reprezintă ce ați fi dorit dacă l-ați folosit numai pentru indentare.

O caracteristică mai periculoasă privind machetarea este reprezentată de atributul `align`, în special când este folosit pentru imagini cu text `alt` (vezi subcapitolul „Grafica" din acest capitol). Pentru un exemplu concret, analizați acest fragment dintr-un articol din știrile CINet:

```
The media has portrayed the outcome of the UPS strike as an important
victory for U.S.trade unions. Certainly, the strikers were successful in
<img.src="/Images/Perspectives/Stories/9_97/per_mission.gif"
width=132 height=121 border=0 align=left hspace=5 vspace=3
alt="Many businesses turned to email for mission critical delivery">
creating a brief moment in which we felt the stalemate, where a number of
people and businesses felt somewhat captive to the rebellion. But overall,
I think the strikers may lose out.
```

Elementul IMG este introdus aici chiar în mijlocul unei propoziții. Când este vizualizată cu un browser grafic, imaginea cu citatul extras din text e deplasată la marginea din stânga, datorită atributului `align=left` (vezi Figura 41.2). Totuși, pentru un browser vocal sau în modul text, alinierea imaginii nu prea are sens, așa că atributul este ignorat. Iată ce auziți că pronunță browserul vocal sau vedeți afișat de un browser în modul text:

```
The media has portrayed the outcome of the UPS strike as an
important victory for U.S.trade unions. Certainly, the strikers
were successful in Many businesses turned to email for mission
critical delivery creating a brief moment in which we felt the
stalemate, where a number of people and businesses felt somewhat
captive to the rebellion. But overall, I think the strikers may
lose out.
```

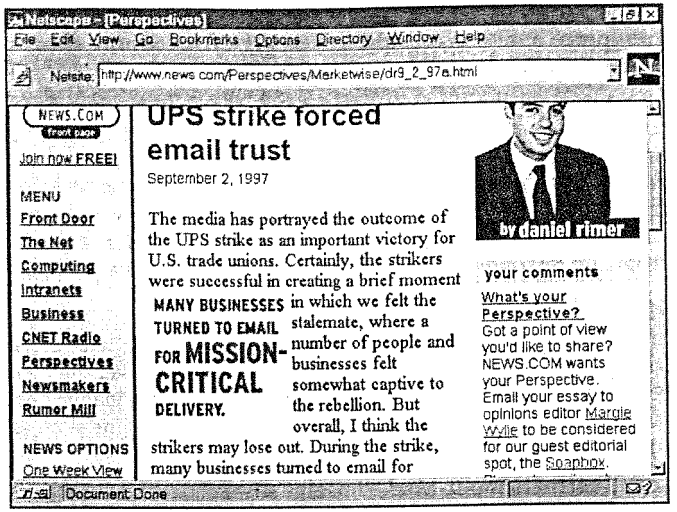



FIGURA 41.2. *Imaginea cu citat extras în articolul din știri, deși folosește în mod corect textul* `alt`*, provoacă dezordine când e redat de un browser în modul text sau un browser vocal.*

După cum vedeți, textul `alt` se intercalează cu corpul textului, făcându-l greu de înțeles. Soluția corectă ar fi evitarea atributului de aliniere a imaginii (de exemplu, folosirea machetării tabelare) sau, ca ultimă soluție, punerea între paranteze a valorii `alt`, pentru a face combinația mai ușor descifrabilă.

Grafica

Folosirea graficii este de departe cea mai obișnuită metodă de a decora paginile dumneavoastră Web. Modelele folosite cuprind de la cele câteva pictograme care decorează o pagină conținând cu preponderență text, până la un sit complet grafic, în care rolul textului este redus la o notă de copyright în subsolul paginii. Restul paginilor Web sunt situate undeva între aceste extreme.

Cele mai multe imagini din Web nu sunt destinate să transmită nici o informație cu adevărat vitală, ci numai să decoreze, să amuze, să expună talentul în proiectare al autorului. Ca urmare, nu trebuie să faceți nici o procesare specială a imaginilor pentru a fi accesate de către persoanele cu handicap – dacă o imagine nu poate fi înlocuită de un șir de text este probabil un element decorativ și poate fi omisă în totalitate, fără probleme (deși foarte rar, există și altfel de situații, când o imagine conține informații importante, care nu pot fi exprimate prin text).

De aceea, cerința principală de accesibilitate pentru folosirea imaginilor este simplă: însoțiți întotdeauna imaginile dumneavoastră cu o reprezentare alternativă prin text (textul `alt`) sau declarați în mod explicit că această imagine nu are altă semnificație decât cea decorativă (pentru aceasta, trebuie folosite textele `alt` goale, după cum se arată în subcapitolul următor). Există totuși unele recomandări referitoare la această problemă, care nu sunt respectate, chiar de către cei care sunt pe deplin conștienți de această cerință de accesibilitatate.

Textele `alt`

Valoarea atributului `alt` din eticheta <IMG> este un șir de caractere definit drept CDATA în definiția DTD din limbajul HTML (vezi Capitolul 4). Această definiție semnifică faptul că aceasta poate conține atât entități mnemonice (cum ar fi ` `) cât și numerice (cum ar fi

 ), dar nu poate conține nici o etichetă de marcare. De exemplu, nu puteți controla întreruperea liniilor în cadrul unui text alt, deoarece orice <BR>: va fi afișat ca atare, în loc de a avea efectul dorit.

Puteți totuși (și chiar este recomandat) să încadrați întreaga imagine în etichetele logice necesare pentru descriere (de exemplu H1, dacă se intenționează ca imaginea să fie titlul principal al paginii). În acest mod, textul alt asociat imaginii va fi afișat sau citit cu voce tare folosind formatarea corespunzătoare (centrarea și evidențierea în modul text, tonul vocii în vorbire ș.a.m.d.). Această convenție nu este respectată totuși de browserele vizuale (pentru mai multe amănunte legate de afișarea textelor alt în browserele vizuale, vezi „Probleme rezultate din practică privind seturile de caractere", din Capitolul 40).

Regulile folosirii textului alt

Care este deci cel mai important principiu de scriere a textelor alt? Potrivit lui Alan Flavell, care a scris un articol amănunțit pe acest subiect (vezi http://ppewww.ph.gla.ac.uk/%7Eflavell/alt/alt-text.html), ar trebui să vă amintiți înainte de toate faptul că textul alt este destinat să comunice informația conținută *în* imagine și nu anumite meta-informații *despre* imaginea în cauză.

În cazul cel mai simplu, când o imagine afișează un șir de text (de exemplu, un titlu sau un buton de pe panoul de navigare), textul alt asociat trebuie să conțină același text și nimic *mai mult*. Nu trebuie să scrieți ceva de genul: „buton mare roșu", „executați clic aici pentru a..." sau, în nici un caz, să prezentați mărimea imaginii în octeți sau pixeli. Asemenea descrieri ar prezenta unele meta-informații privitoare la imagine și rolul acesteia pe pagină și deoarece aceste informații lipsesc din imaginea propriu-zisă, ele sunt redundante în textul alt asociat imaginii. Pentru un exemplu privind folosirea textului alt în mod corespunzător (deși formatarea este incorectă), consultați în subcapitolul precedent „Alte instrumente de machetare" din acest capitol, imaginea ecranului CINet; a scrie ceva de genul alt="pull-out quote" ar fi o greșeală evidentă în acest caz.

Bineînțeles că nu este întotdeauna la fel de simplu. Pentru emblema unei companii, de exemplu, nu este chiar așa de ușor să decidem ce ne spune imaginea. Ne dăm seama întotdeauna că este o emblemă sau o percepem ca un mod decorativ de afișare a numelui companiei? Cred că de cele mai multe ori emblemele sunt folosite în paginile Web drept simple anteturi, așa că este mai bine să spunem alt=" Foo Inc." decât alt=" Logo of Foo Inc." . Uneori aveți nevoie totuși să accentuați faptul că imaginea reprezintă o emblemă (de exemplu, într-o colecție de embleme online pentru proiectare), așa încât a doua variantă a textului alt poate fi de folos.

În următoarea listă, am prezentat pe scurt regulile de bază pentru alegerea textelor alt în cazul diferitelor tipuri de imagini:

- **Spații și decorații:** Pentru imagini cu importanță mică sau fără importanță pentru semnificația paginii, *întotdeauna* precizați un text alt gol (alt=" ")

S-ar putea să gândiți că aceasta este oricum valoarea implicită, dar nu este așa. Specificația HTML 4.0 prezintă un număr de surse pe care browserul ar trebui să le folosească pentru a construi un text alt, pentru o imagine fără atributul alt, iar rezultatul cel mai probabil al algoritmului este numele fișierului cu imaginea (fără extensie). Cel mai cunoscut browser în modul text, Lynx,, folosește [IMAGE] ca înlocuitor pentru imaginile fără atributul alt. Așa că alt=" " reprezintă o necesitate dacă nu doriți ca cititorii dumneavoastră să devină exasperați de tot felul de lucruri nefolositoare.

- **Marcatori de listă:** Imaginile de acest gen pot fi considerate drept decorative, dar de fapt ele nu sunt așa; ele folosesc unui scop foarte practic și anume acela de a separa elementele unei liste. De aceea, șirul alt ar trebui să prezinte ceea ce este folosit în mod convențional ca marcator de listă în documentele text – cel mai adesea caracterul *.

Considerând că ceea ce s-a spus despre liste (vezi subcapitolul „Liste", prezentat anterior în acest capitol), este o idee bună să se pună numere sau litere consecutive ca texte alt pentru un rând de marcatori grafici de listă. În acest fel, nu distrageți atenția utilizatorilor browserelor vizuale cu o numărătoare redundantă, deși furnizați un mijloc de navigare pentru utilizatorii cu handicap.

- **Linii despărțitoare:** Din nou, un element decorativ, care nu îndeplinește niciodată o funcție utilă; înlocuitorii ASCII simpli pentru liniile despărțitoare grafice pot fi =====, -----, ***** ș.a.m.d. Observați că această recomandare intră în conflict cu avertizările privind utilizarea regulilor neuzuale de punctuație (vezi observația din subcapitolul „Marcarea și formatarea textului", prezentat mai devreme în acest capitol); închipuiți-vă că trebuie să ascultați șirul fără sfârșit :„linie de despărțire - linie de despărțire - linie de despărțire...", mormăit de un browser vocal, în locul unei linii grafice de despărțire. În ciuda acestui lucru, liniile grafice de despărțire sunt preferate elementelor HR în documentul cu „Regulile unitare de accesibilitate" tocmai deoarece permit stabilirea valorilor alt.

Mai simplu spus, un text alt ideal ar trebui să fie astfel încât utilizatorii de browsere în modul text sau vocal să nu își dea seama că înlocuiește o imagine, dar să-l perceapă ca parte integrantă din textul paginii.

Prezentarea textelor alt

După cum ați văzut în exemplul CINet, din subcapitolul „Alte instrumente de machetare", browserele ne-grafice nu tratează textele alt într-un mod diferit față de textul obișnuit din pagină; nu sunt adăugați separatori, iar textul alt este pur și simplu afișat în locul unei imagini. De exemplu, dacă două imagini sunt puse una lângă alta, textele lor alt asociate nu vor fi separate deloc prin spațiu. Acest lucru poate afecta semnificativ o bară de navigare realizată dintr-un rând de imagini alăturate (pentru mai multe detalii, vezi subcapitolul „Să spunem nu hărților imagine", din acest capitol).

Prin urmare, în cele mai multe cazuri este recomandabil să se introducă unele mijloace de separare chiar în textul alt. Puteți să adăugați un spațiu la sfârșitul textului alt, după cum se vede mai jos:

```
alt="text "
```

Această soluție este funcțională, dar nu este elegantă. Cea mai sigură cale este să se pună textul între paranteze:

```
alt="[text]"
```

Puteți adăuga de asemenea un caracter ¦, după cum urmează (această variantă poate fi folosită pentru a separa textele alt asociate mai multor imagini dintr-un rând):

```
alt=" text ¦"
```

Browserele vizuale pun o altă problemă. Dacă ați specificat valorile height și width corespunzătoare unei imagini (și ar trebui să faceți acest lucru deoarece această măsură îmbunătățește în mod spectaculos viteza de descărcare și afișare a paginii), textul alt asociat s-ar putea să nu intre în spațiul alocat. Pentru utilizatorii care au deconectat încărcarea automată a imaginilor, aceste texte alt nu ar putea fi citite. Ca un compromis, puteți scoate atributele height și width pentru acele imagini care nu afectează în mod

semnificativ viteza de descărcare a paginilor; acest lucru va cere browserului să aloce un dreptunghi destul de mare pentru fiecare imagine, suficient să cuprindă textul `alt`.

Alternative pentru textele `alt`

Utilizarea atributului `alt` cu eticheta `IMG` este larg răspândită și general acceptată, dar are unele limitări serioase. Faptul că textele nu pot conține marcări HTML și (din teama de a nu întrerupe afișarea în cazul unor anumite browsere) nu pot depăși o anumită lungime ar putea constitui un obstacol pentru cazul în care o imagine (cum ar fi o diagramă științifică) conține foarte multe date, majoritatea acestora putând fi comunicate sub formă de text utilizatorilor care nu pot vizualiza grafica.

În astfel de cazuri, soluția disponibilă este să se realizeze o subpagină separată pentru fiecare imagine, care să conțină notele explicative. Problema care rămâne totuși este cum să se lege această pagină de pagina principală astfel încât să fie evident pentru utilizatorii browserelor ne-grafice și să nu distragă atenția pentru utilizatorii celor grafice. O metodă sugerată este să se folosească litera mare „*D*" (de la „descriere") pentru legătură, poziționată lângă colțul din dreapta-sus al imaginii.

O soluție mult mai elegantă este folosirea unei imagini mici și transparente pentru legătură, care este invizibilă pentru browserele grafice, situată lângă imaginea care este explicată și alocarea unui text `alt` care să facă evident scopul acesteia pentru utilizatorii browserelor ne-grafice.

Pe viitor, vor fi disponibile și alte opțiuni de explicare a imaginilor. Noul format grafic numit PNG (Portable Network Graphics), destinat să înlocuiască eventual formatul GIF, prevede păstrarea șirurilor arbitrare de text chiar în fișierul imagine. Specificația HTML 4.0 sugerează ca eticheta `IMG` să fie înlocuită în timp de elementul `OBJECT`, a cărui funcție este similară textului `alt`:

```
<OBJECT data="TheEarth.gif">
 The <STRONG>Earth</STRONG> as seen from space.
</OBJECT>
```

Șirul dintre `<OBJECT>` și `</OBJECT>` va fi redat numai prin acei agenți pentru utilizatori care nu pot afișa imaginea. În acest mod, orice text cu orice marcare poate fi folosit ca o alternativă de prezentare pentru o imagine.

Să spunem nu hărților imagine

Hărțile imagine au fost popularizate cândva ca o îmbunătățire semnificativă a schemei originale HTML o-legătură-pentru-fiecare-element. Capacitatea de a asocia adrese URL la fiecare domeniu al unei imagini a fost într-adevăr interesantă în perioada în care imaginile erau percepute ca ființe extraterestre împrăștiate pe paginile sărace în grafică ale browserului Mozaic. De atunci, instrumentele de machetare (sau artificiile?) s-au îmbunătățit totuși considerabil și pentru majoritatea paginilor proiectate serios, imaginile au devenit baza compoziției paginii.

Browserele grafice moderne sunt capabile să poziționeze imaginile cu o precizie de până la un pixel (chiar fără foi de stil, o astfel de poziționare precisă poate fi îndeplinită folosind tabelele și imaginile pentru spațiere). Această caracteristică face de fapt inutile hărțile imagine: puteți compartimenta panoul de navigare în butoane separate cu numai o legătură pentru fiecare imagine și le puteți aranja oriunde în machetă. Ar trebui să fiți totuși reținut în a vă angaja în această muncă manuală fără a avea suficiente motive. Care sunt, în orice caz, dezavantajele folosirii hărților imagine?

După cum ați putea ghici, cel mai important dezavantaj al acestora este accesibilitatea scăzută pentru agenții de utilizator ne-grafici. Acest lucru este adevărat în mod special pentru hărțile imagine de pe server care necesită trimiterea unei perechi de coordonate către server. Browserele în modul text și cele vocale nu pot în nici un fel să înțeleagă semnificația nici unei valori a coordonatelor, alta decât să încerce perechea „0,0" (de aceea este recomandat să se lege acest punct de origine de o pagină alternativă care conține numai text).

Hărțile imagine dinspre client, care folosesc elementele `MAP` și `AREA`, sunt mult mai prietenoase pentru persoanele cu handicap, deoarece fișierul HTML conține toate informațiile privitoare la legăturile hărții imagine, precum și adresele lor URL. Textul `alt` pentru fiecare legătură este tot ceea ce lipsește unui browser ne-grafic pentru a fi capabil să construiască o alternativă viabilă pentru o hartă imagine. Este de ajuns să se afișeze (sau să se citească) rândul de texte `alt` ca legături și utilizatorul nu va suspecta niciodată că sunt extrase dintr-o hartă imagine.

Acest lucru nu este o deficiență a standardului: atributul `alt` este într-adevăr destinat elementului `AREA` și, în plus, este singurul atribut obligatoriu. Următorul exemplu prezintă modul de folosire:

```
<IMG src="navbar.gif" USEMAP="#navmap">

<MAP NAME="navmap">
  <AREA coords="0,0,50,50"
    href="page1.html" ALT="Previous Page">
  <AREA coords="51,0,100,50"
    href="/"          ALT="Home">
  <AREA coords="101,0,150,50"
    href="page3.html" ALT="Next Page">
</MAP>
```

Totuși suportul pentru asemenea hărți imagine, care devin cu timpul demodate, este departe de a fi universal. Chiar și autorii care sunt conștienți de necesitatea stabilirii textelor `alt` pentru imagini, uită adesea să includă textele `alt` în elementele `AREA` ale hărții imagine. Mai rău chiar, browserele ne-grafice sunt departe de a fi toate suficient de inteligente pentru a recunoaște textele `alt` din zonele `AREA` și să le prezinte pe acestea drept legături. Același lucru este adevărat când este vorba despre browserele grafice obișnuite având dezactivată încărcarea automată a imaginilor.

De aceea, principalul motiv pentru abandonarea hărților imagine este accesibilitatea: când fiecare legătură este conectată cu propriul element `IMG` și cu propriul text `alt`, atunci există garanția ca această legătură să lucreze cu toți agenții pentru utilizator sau cu orice metodă de acces. Sunt totuși mai multe motive pentru care abordarea de tipul o-legătură-pentru-o-imagine este de preferat:

- Dacă alte pagini ale sitului folosesc panouri de navigare similare, dar nu identice, ați putea refolosi butoanele păstrate în fișiere grafice separate, reducând astfel timpul de descărcare total al sitului.
- Folosind fișierele GIF care sunt limitate la un maxim de 256 de culori, secționarea unei bare de navigare în bucăți permite fiecărui buton să aibă propria paletă separată, îmbunătățind astfel redarea culorii pentru sistemele cu rezoluție mare de culoare sau cu culoare reală (sistemele care nu intercalează toate imaginile folosind o paletă fixă).

Formulare și legături

În principiu, un formular HTML este un dispozitiv pentru colectarea intrărilor de la utilizator și trimiterea acestora la server. Deoarece datele de intrare sunt aproape întot deauna sub formă de text, acest proces este puțin probabil să pună probleme de accesibilitate, cu condiția ca utilizatorii cu handicap să posede mijloace convenabile pentru ca să navigheze prin formulare și să interacționeze cu elementele formularului fără a folosi mouse-ul.

În HTML 4.0, au fost introduse unele elemente și atribute noi pentru a realiza o marcare mai cuprinzătoare și ca să îmbunătățească accesibilitatea prin tastatură a formularelor. Unele dintre noile atribute afectează legăturile hipertext precum și elementele formularelor. Marea problemă cu toate aceste îmbunătățiri este totuși că nu toate browserele le acceptă încă.

Elementele LABEL, FIELDSET și LEGEND

Noul element LABEL asociază un fragment din conținutul unui formular cu unul din controalele formularului (vezi Capitolul 15, „Crearea și utilizarea formularelor HTML"). Astfel, se realizează funcționalitatea care lipsea formularelor HTML în comparație cu casetele de dialog din Windows, de exemplu, unde puteți executa clic pe șirul de text de lângă un buton radio pentru a-l activa sau dezactiva. Dacă marcați acum un fragment de text sau alt conținut ca fiind LABEL pentru controlul formularului, această etichetă va transmite focusul primit (prin mouse, tastatură sau alte mijloace) la controlul corespunzător.

În consecință, importanța accesibilității elementului LABEL este aceea că furnizează un fel de „text alt" vizibil pentru controlul cu care este asociat. Elementul INPUT (dar nu și celelalte controale ale formularului) s-ar putea să aibă și el un atribut alt propriu; este de dorit totuși ca aceeași valoare a etichetei să fie accesibilă pentru toți utilizatorii formularului, nu numai pentru cei care folosesc accesul vocal sau browsere care nu acceptă formularele (care se presupune că prezintă șirul alt al unui element INPUT).

Ceea ce face LABEL pentru controalele independente, pot face elementele FIELDSET și LEGEND pentru un grup de controale. Prin includerea unui număr de controale dintr-un formular într-un element FIELDSET, le integrați într-un grup care e tratat într-o oarecare măsură ca un întreg. Elementul LEGEND, care este primul element obligatoriu al elementului FIELDSET, înconjoară un fragment de text care servește ca etichetă pentru întreg grupul.

Navigarea cu ajutorul tastaturii

Fiind un dispozitiv mai adecvat persoanelor cu handicap decât mouse-ul, tastatura este importantă, deoarece în primul rând toți utilizatorii posedă mijloace convenabile de navigare cu tastatura prin intermediul „punctelor fierbinți" din pagină și, în al doilea rând, autorii de pagini Web pot redefini secvențele de acces pentru paginile lor în vederea obținerii eficienței maxime. Aceste „puncte fierbinți" cuprind controalele de introducere a datelor în formular și legăturile hipertext. Observați că chiar dacă accentul cade pe o legătură, puteți totuși să dați comenzi suplimentare (cum ar fi apăsarea tastei ENTER) pentru activarea acesteia.

Cele două moduri de acces prin tastatură furnizate de limbajul HTML 4.0 sunt tastele de acces, sau scurtăturile, și ordinea de tabulare. *Tastele de acces* sunt definite drept caractere care, în momentul în care sunt apăsate pe tastatură simultan cu o anumită tastă modificatoare (Alt la sistemul Windows, Cmd la Macintosh), mută focusul pe elementul asociat cu această tastă de acces. *Ordinea de tabulare* definește ordinea în care elementele primesc focusul când o anumită tastă de tabulare (de obicei tasta Tab) este apăsată în mod repetat.

Elementele HTML diferă prin modurile de acces de la tastatură cu care pot fi folosite. Stabilirea poziției ordinii de tabulare este posibilă pentru cele patru controale de intrare (INPUT, SELECT, TEXTAREA și BUTTON) ca și pentru elementele A, AREA și OBJECT. Tastele de acces pot fi, pe de altă parte, asociate numai elementelor LABEL, LEGEND și A.

Se pot folosi astfel scurtături pentru a se face comutarea la elemente de control etichetate prin LABEL, la grupuri de elemente de control din interiorul elementelor FIELDSET și la legături independente, deși tabularea va lucra pentru toate elementele de control ale formularelor (nu contează dacă sunt asociate cu elementele LABEL sau FIELDSET sau nu) și pentru hărțile imagine, obiectele OBJECT, precum și legăturile independente.

Pentru a specifica pozițiile pentru ordinea tabulării unui element, folosiți atributul tabindex, după cum se vede în acest exemplu:

```
<FORM action="..." method="post">
<INPUT tabindex="1" type="text" name="field1">
<INPUT tabindex="2" type="text" name="field2">
<INPUT tabindex="3" type="submit" name="submit">
</FORM>
```

Iată algoritmul complet care, potrivit specificației HTML 4.0, ar trebui urmat de către agenții pentru utilizator pentru a defini ordinea de tabulare într-o pagină:

1. Acele elemente care acceptă atributul tabindex și atribuie o valoare pozitivă acestuia sunt navigate mai întâi. Navigarea începe de la elementul cu cea mai mică valoare tabindex și se termină cu elementul cu cea mai mare valoare. Valorile nu trebuie să fie succesive sau să înceapă cu o anumită valoare. Elementele care au valori identice pentru tabindex ar trebui să fie navigate în ordinea în care apar în document.

2. Următoarele elemente navigate sunt acele elemente care nu definesc valoarea atributului tabindex sau nu o acceptă. Aceste elemente sunt navigate în ordinea în care apar în document.

3. Elementele care acceptă atributul tabindex și îi atribuie o valoare negativă nu participă la ordinea de tabulare.

4. Elementele dezactivate nu participă la ordinea de tabulare.

Tastele de acces sunt atribuite prin folosirea atributului accesskey, după cum urmează:

```
<LABEL for="field1" accesskey="A">Enter Your <U>A</U>ge:</LABEL>
<INPUT type="text" id="field1">
```

Valoarea atributului accesskey trebuie să fie un singur caracter (indiferent de tipul literelor) din codificarea caracterelor de document (vezi Capitolul 40).

Alte modalități

În acest ultim subcapitol, voi lua în discuție aspectele privind accesibilitatea celorlalte elemente ale paginii Web care nu au fost prezentate în subcapitolele precedente referitoare la „Text", „Grafică", „Machetare" și „Formulare".

Cadrele

Documentele care folosesc cadrele sunt o adevărată calamitate pentru utilizatorii cu handicap (și pentru motoarele de căutare, după cum veți vedea în capitolul următor). Ele nu sunt în principiu inaccesibile deoarece folosesc un mediu neconvențional sau altceva; de fapt, pagina rădăcină a unui set de cadre nu este altceva decât o colecție de legături cu paginile cadru, deși aceste legături sunt exprimate folosind etichetele FRAME în locul ancorelor hipertext convenționale.

Bineînțeles, este aproape imposibil pentru un orb să *folosească* într-adevăr o pagină cu cadre – mai dificil poate decât a juca șah fără să te uiți pe tablă. În ciuda acestui lucru, este evident că paginile cu cadre pot fi utilizate într-o *oarecare* măsură permițând urmărirea referințelor etichetei FRAME de către persoanele handicapate, ca și cum ar fi legături hipertext obișnuite. Din păcate, nici un agent utilizator nu implementează în prezent această funcție, după câte cunosc.

Pentru cititoarele de ecrane, o pagină cu cadre este un exemplu exacerbat al machetării tabelare (vezi subcapitolul „Tabelele", prezentat anterior în acest capitol) cu tot noianul de consecințe nedorite. Alte cititoare de ecrane încearcă să trateze fiecare cadru ca o fereastră separată, încât uneori s-ar putea să fie dificil să se comute programul pe alt cadru.

Există totuși o metodă de a face siturile cu cadre accesibile persoanelor cu handicap: folosirea etichetei `NOFRAMES`. Acest procedeu este prezentat în detaliu în Capitolul 42.

Audio și video

Asemeni imaginilor, clipurile audio și video sunt în mod inerent inaccesibile unor categorii de utilizatori. Metodele de rezolvare a acestei probleme sunt similare celor puse la punct pentru imagini (vezi subcapitolul „Grafica", prezentat anterior în acest capitol).

Dacă un clip audio sau video este purtătorul unor informații importante pe care doriți să le acceseze toată lumea, el ar trebui să fie însoțit de transcrierea acestuia (pentru citirea lor vorbită, conversații ș.a.m.d.) sau de o descriere generală (pentru sunetele neverbale sau scene). Ca în cazul textelor `alt` pentru imagini, ar trebui să transmiteți informația conținută *în* clip și nu meta-informații *despre* clipul respectiv. De exemplu, indicarea mărimii fișierului pentru fișierele audio sau video poate fi utilă, dar nu-și are locul în cadrul reprezentării alternative a clipului multimedia; această informație poate fi necesară celor care *pot* vedea/auzi prin multimedia, nu celor care sunt lipsiți de această posibilitate.

Din punct de vedere tehnic, transcrierea sau descrierea poate fi plasată pe o pagină separată legată de clipul căruia îi aparține. De asemenea, deoarece aceste tipuri de date sunt probabil încapsulate folosind elementul `OBJECT`, puteți folosi posibilitatea alternativă de redare: tot conținutul aflat între `<OBJECT>` și `</OBJECT>` va fi redat numai de acei agenți pentru utilizator care nu pot reda elementul `OBJECT` propriu-zis.

Java și JavaScript

Aceste două limbje de programare pentru Web sunt adesea considerate ca fiind ceva din lumea super-tehnologiei browserelor grafice de ultimă oră, cu capacitate nelimitată și interfețe sofisticate. Este adevărat că numai principalele browsere grafice din prezent acceptă Java și JavaScript, dar acest lucru nu înseamnă că trebuie să le abandonați complet pentru ca paginile dumneavoastră să fie universal accesibile.

Java nu este un instrument care să producă abțibilduri pentru paginile Web; este un limbaj de programare complet dezvoltat cu un potențial ridicat pentru construirea aplicațiilor eficiente și portabile de procesare a informațiilor. Appleturile Java pot fi folosite – și *sunt* folosite – pentru o mulțime de operații care nu pot fi realizate cu orice altă tehnologie. Sunt situații în care nu se pune problema dacă este practic sau nu să se folosească Java; se poate pune însă problema privind modul în care pot fi făcute accesibile aplicațiile Java.

Principiile de bază (modalitățile alternative, descrierea mai mult logică decât fizică a datelor) rămân valabile, dar există bineînțeles destule recomandări specifice limbajului Java. Accesibilitatea limbajului Java este un subiect vast, iar o carte HTML nu este probabil locul potrivit să se discute acest lucru în detaliu. Pentru a afla ultimele evoluții în ceea ce privește accesibilitatea prin Java și JavaScript, consultați situl întreținut de Trace R&D Center la `http://www.trace.wisc.edu/world/java/java.htm`.

O observație finală

O soluție de accesibilitate destul de populară pe care nu am menționat-o până acum în acest capitol este o versiune text integrală a întregului sit sau pagină, legată undeva aproape de partea de sus a paginii „reale", ca o ieșire de urgență pentru cei care nu pot suporta ceva mai complicat decât HTML 2.0. Aceasta nu este o omisiune; sincer, am lăsat la urmă această posibilitate intenționat. După cum puteți concluziona din acest capitol lung și diversificat, sunt foarte puține situațiile care pun probleme grele de accesibilitate. Până la urmă, limbajul HTML a fost destinat chiar de la început să fie accesibil, portabil și un mediu ușor de transformat și nici o „extensie" nu poate să-l deposedeze de caracteristica sa esențială.

În marea majoritate a cazurilor, paginile Web inaccesibile nu sunt rezultatul tehnologiilor aplicate, ci al implementării incorecte și al neatenției. Configurarea unei versiuni text integrale este nu numai ineficientă și predispusă la erori și desincronizări; este, înainte de toate, o predare, o capitulare, o trădare a spiritului HTML. Din contră, folosirea apreciabilă și accesibilă a întregului inventar de instrumente HTML este o favoare pentru cei care întreține situl, utilizatori și în final, chiar pentru limbaj.

Rezumat

În acest capitol, am discutat despre metodele de a face documentele HTML mai accesibile pentru persoanele cu handicap fizic sau senzorial, cuprinzând atât procedeele verificate cât și cele mai recente dezvoltări ale limbajului HTML 4.0. Cele mai multe din recomandările din acest domeniu pot fi rezultatul a două principii de bază ale accesibilității: prezentarea informației în modalitățile alternative și marcarea logică în detrimentul celei reprezentative.

Au fost discutate următoarele aspecte referitoare la scrierea în HTML:

- **Text:** Ce etichete trebuie evitate în marcarea textului, care sunt sfaturile importante pentru selectarea culorilor și prezentarea listelor
- **Machetare:** Cum se folosesc corespunzător tabelele și alte instrumente de machetare
- **Grafică:** De ce sunt importante textele `alt`, care sunt regulile de bază pentru compunerea textelor `alt` utile și de ce trebuie evitate hărțile imagine
- **Formulare:** Ce mijloace noi de navigare cu ajutorul tastaturii sunt furnizate în HTML 4.0 pentru formulare și cum se folosesc acestea
- **Diverse:** Care sunt considerentele de accesibilitate pentru folosirea cadrelor, clipurilor multimedia, a limbajelor Java și JavaScript

Capitolul 42
Strategii pentru indexare și motoare de căutare

de Dmitry Kirsanov

În acest capitol:

- **Cum funcționează motoarele de căutare 660**
- **Cum se face proiectarea motoarelor de căutare 665**

Unii webmasteri pretind că mai mult de jumătate din traficul total din siturile lor provine de la motoarele de căutare. Acest procent va depinde bineînțeles de conținutul sitului dumneavoastră și de publicul așteptat, dar importanța acestui instrument gratuit și eficient de publicitate în Web nu poate fi negată.

Trebuie să admitem că traficul generat de motoarele de căutare conține o mulțime de „resturi" – adică vizite inutile din partea persoanelor care au fost îndreptate în mod greșit către situl dumneavoastră prin „cuvinte cheie ghicite", în căutarea unui lucru total diferit. Totuși, acei vizitatori care sfârșesc prin a găsi ceea ce caută reprezintă o categorie foarte valoroasă – poate cea mai valoroasă din cadrul publicului dumneavoastră.

Poate că nu ar fi aflat niciodată despre situl dumneavoastră din oricare altă sursă și găsindu-l singuri, fără nici o reclamă sau indicație, este mult mai probabil să fie satisfăcuți de descoperire. De fapt, motoarele de căutare reprezintă aproximarea cea mai apropiată posibil de idealul de distribuție a informațiilor în mod gratuit și independent: dumneavoastră căutați lucrurile de care aveți nevoie și obțineți lucrurile căutate, fără nici o influență de marketing sau politică.

Bineînțeles, acest ideal nu este ușor de atins. Trebuie să învățați mai întâi despre unele procedee pentru a atrage și a primi vizitatori în situl dumneavoastră, despre păianjeni automați care indexează paginile dumneavoastră pentru motoarele de căutare și în al doilea rând despre utilizatorii motoarelor de căutare care ar putea fi interesați de conținutul sitului dumneavoastră (aceste probleme sunt puțin diferite, deși sunt legate una de alta).

Acest capitol conține două subcapitole principale. În primul, veți face cunoștință cu modul în care lucrează motoarele de căutare și care sunt caracteristicile principale ale celor mai importante motoare aflate în prezent în exploatare. În al doilea subcapitol, voi aplica aceste cunoștințe pentru a contura un set de recomandări specfice pentru un proiect HTML de căutare eficient și prietenos.

Cum funcționează motoarele de căutare

În primul rând, am vești proaste. Când veți investiga domeniul, veți descoperi că motoarele de căutare sunt, înainte de toate, o tehnologie proprie pe o piață foarte competitivă. Mai simplu spus, companiile care produc motoare de căutare păstrează bine secretele acestora și dezvăluie publicului numai ce consideră ele de cuviință.

Datorită importanței – și necesității – atenției speciale și dotării sitului dumneavoastră în ceea ce privește motoarele de căutare, acest comportament este într-adevăr descurajant. Webmasterii care sunt interesați de problemă trebuie să se bazeze în special pe propriile

cercetări, care sunt adesea deviate de la subiect și incomplete. Zvonurile, bârfa și controversele sunt răspândite.

Bineînțeles că există situri ale căror persoane care realizează întreținerea sunt ocupate cu colectarea informației și efectuarea cercetărilor în acest domeniu (cum ar fi de exemplu Search Engine Watch, la adresa `http://searchenginewatch.com`), așa că acest capitol a extras o mare parte din material din aceste surse utile. Când este vorba totuși despre detaliile tehnologiei motoarelor de căutare, ar trebui să considerați concluziile acestor terți investigatori precum un grăunte de nisip.

Pe de altă parte, numai cercetările pot da răspunsurile la anumite întrebări, deoarece sistemele de la acest nivel de complexitate tind să fie guvernate de statistica factorilor multipli mai degrabă decât de logica simplă. De fapt, unele particularități în comportarea fiarelor de căutare ar putea fi o surpriză la fel de mare pentru creatorii lor, cât și pentru marele public. Sistemele de procesare a unei cantități enorme de informații se comportă uneori ca ființe umane și nu ca mașini neînsuflețite.

Cu aceste restricții în minte, să luăm în considerare principalele angrenaje care se învârt în interiorul unui motor obișnuit de căutare.

Păianjeni

Păianjenii de indexare (uneori numiți roboți – robots, sau tăuni – bots sau târâtori – crawlers) sunt agenții secreți care fac munca al cărui rezultat îl savurați când faceți căutarea. Programele păianjen, ca și browserele, cer și extrag documente din serverele Web, dar spre deosebire de browsere, ei fac acest lucru nu pentru a fi văzute de oameni, ci pentru indexarea automată și includerea lor în baza de date proprie. Ei fac acest lucru fără încetare, în cantități aproape de neimaginat (milioane de pagini pe zi), 24 de ore din 24 și fără zile libere.

Păianjenii reprezintă ceea ce deosebesc motoarele de căutare de *directoare* (unul din cele mai cunoscute directoare este Yahoo!; vezi `http://www.yahoo.com`). Directoarele nu au păianjeni favoriți deoarece toate legăturile într-un director sunt descoperite (sau oferite), examinate și marcate de oameni. Această diferență face ca resursele culese cu mâna ale directoarelor să fie, în medie, mult mai valoroase, dar mult mai puțin voluminoase față de mulțimea omogenă de legături dintr-un motor de căutare.

Fiecare document nou întâlnit de un păianjen este scanat căutându-se legăturile, iar aceste legături sunt fie imediat traversate, fie programate pentru regăsirea ulterioară. Teoretic, prin urmărirea tuturor legăturilor care pornesc dintr-un set inițial de documente reprezentative, un păianjen va termina prin a indexa întreaga rețea Web.

Totuși, acest scop nu poate fi atins în practică. Pentru început, o mulțime de documente din Web sunt generate în mod dinamic, cel mai adesea ca răspuns la o intrare dintr-un formular. În mod obișnuit, deși păianjenii pot urmări legăturile, ei nu știu ce să pună în câmpurile unui formular, așa că orice dată regăsită la cerere este inaccesibilă pentru păianjenii de căutare (dacă nu e furnizat alt mecanism alternativ de acces). Acestei categorii îi aparțin diferite baze de date accesibile din Web, inclusiv motoarele de căutare.

Păianjenii nu pot de asemenea accesa paginile care sunt personalizate cu ajutorul blocurilor cookie sau a paginilor folosind orice artificiu JavaScript sau Java care le modifică conținutul. Unii păianjeni nu înțeleg nici măcar cadrele. (vezi Subcapitolul „Cadrele", din acest capitol). Cum probabil ați ghicit, motoarele de căutare nu pot să facă deocamdată capete sau cozi nici unei imagini sau clip audio sau video, așa că acești biți de informație sunt irosiți (de fapt ei nu sunt nici măcar ceruți de păianjeni). Ce rămâne este sursa HTML

pură, din care păianjenii desprind suplimentar toate marcările și etichetele pentru a ajunge la scheletul reprezentat de textul simplu.

Chiar și cu aceste supoziții economice, împachetarea întregului Web într-o singură bază de date se dovedește a fi o sarcină practic imposibilă. Ar fi fost posibilă acum un an, dar nu acum când rețeaua Web a devenit *așa* de mare. De aceea motoarele de căutare evoluează de la strategia de a înghiți tot ce văd către diferite procedee de selecție.

În mod ideal, această selecție ar trebui să aibă drept scop îmbunătățirea calității bazei de date prin abandonarea resturilor și scanarea numai a conținutului primar al sitului Web. Bineînțeles că în realitate acest tip de discernământ este imposibil deoarece nu există nici un program automat care să fie suficient de inteligent să separe grâul de neghină. Singura cale de sortare este introducerea unor restricții arbitrare.

Un motor de căutare care acceptă „păianjeni cu eșantionare" este AltaVista (`http://altavista.digital.com`). Se spune că norma pentru păianjenul AltaVista este nu mai mult de 600 de documente pe oricare domeniu. Dacă este adevărat, acest lucru înseamnă că domenii mari precum `geocities.com` sau chiar `microsoft.com` sunt subreprezentate în mod sever în baza de date AltaVista. Rămâne deschisă speculațiilor întrebarea dacă alte motoare de căutare folosesc tehnici de eșantionare analoage sau mărimea bazelor lor de date e limitată numai de capacitatea lor tehnică.

Toate motoarele de căutare permit utilizatorilor să adauge adresele lor URL în baza de date pentru folosirea păianjenului. Unele regăsesc documentele prezentate imediat, iar altele le programează pentru scanările viitoare, dar în orice caz, acest lucru vă permite să fiți sigur că domeniul dumneavoastră nu lipsește. Vi se cere să introduceți numai adresa rădăcină URL a sitului dumneavoastră, iar folosirea acestui mecanism pentru înregistrarea fiecărei pagini a fost considerată ca fiind un fel de spamming (distribuirea unor prezentări, fără ca acestea să fie solicitate). Pe de altă parte, fiind cunoscută natura selectivă a păianjenului, nu este o idee rea să se înregistreze cel puțin toate paginile importante ale sitului dumneavoastră (aveți totuși grijă: unele motoare de căutare limitează numărul de prezentări pe domeniu).

O altă problemă importantă este cât de des actualizează păianjenii bazele lor de date prin revizitarea siturilor pe care le-au indexat deja. Acest parametru diferă semnificativ de la un motor de căutare la altul, cu perioadele de actualizare stabilite de la o săptămână și până la câteva luni. Acest aspect al performanțelor motoarelor de căutare permite o oarecare estimare independentă: puteți analiza jurnalul de acces al serverului pentru a vedea când ați fost vizitat de păianjeni și ce document au cerut. În acest scop este disponibil un script de mare ajutor al firmei Perl, numit BotWatch, la adresa `http://www.tardis.ed.ac.uk/~sxw/robots/botwatch.html`.

Multe motoare de căutare au probleme cu siturile când se folosesc alte limbi decât engleza, în special dacă setul de caractere este diferit de ISO 8859-1 (vezi Capitolul 40, „Internaționalizarea limbajului HTML"). De exemplu, HotBot (`http://www.hotbot.com`) nu returnează nimic în cazul în care este interogat cu cuvinte cheie în limba rusă și nu poate nici să afișeze rezumatul documentelor în limba rusă. Acest lucru îl face inutil pentru vizitatorii ruși, în ciuda faptului că păianjenul HotBot scanează curent o bună parte a siturilor Web din Rusia.

Interfața de căutare

Interfața de căutare este partea vizibilă a aisbergului reprezentat de motorul de căutare. În fiecare zi, milioane de oameni introduc un număr indefinit de cuvinte cheie în formularele de căutare și obțin un număr indefinit de adrese URL ca răspuns. Aceasta este deja una dintre cele mai mari și mai intens folosite resurse de informații de pe glob.

Nu vă voi învăța *cum* se folosesc motoarele de căutare, deoarece nu acesta este scopul acestei cărți. Totuși, pentru a crea documente HTML prietenoase la căutare, trebuie să țineți cont de gama de caracteristici oferite utilizatorilor de motoare moderne de căutare.

Opțiuni de bază

Toate motoarele de căutare importante au, pe lângă forma cea mai simplă de interogare cu unul sau mai multe cuvinte cheie, și unele opțiuni suplimentare de căutare. Totuși, scopul acestor caracteristici diferă semnificativ și nu este stabilită încă nici o sintaxă standard pentru invocarea lor. Printre cele mai obișnuite opțiuni de căutare se află:

- **Operatori booleeni:** `AND` (găsește tot), `OR` (găsește oricare), `AND NOT` (excludere), pentru combinarea cuvintelor cheie în interogări
- **Căutarea după frază:** Caută cuvintele cheie numai dacă se află unul lângă altul în document, în această ordine
- **Proximitate:** Caută cuvintele cheie dacă sunt suficient de apropiate unul de altul (noțiunea „suficient de aproape" începe de la două cuvinte intercalate pentru WebCrawler și până la 25 de cuvinte, în cazul Lycos)
- **Căutare după mediu:** Caută paginile care conțin appleturi Java, obiecte Shockwave ș.a.m.d.
- **Căutări speciale:** Caută cuvinte cheie sau adrese URL din cadrul legăturilor, numelui imaginilor, titlurilor de documente
- **Restricții de căutare:** Limitează căutarea la un interval de timp în care a fost creat documentul, specificând o limbă pentru document (Alta Vista) ș.a.m.d.

Chiar dacă aveți un inventar complet de asemenea zorzoane, nu vă puteți aștepta ca motorul de căutare să aibă aceleași performanțe cu, să spunem, dialogul Search din Microsoft Word.

De exemplu, AltaVista sugerează folosirea bazei de date pe care o conține ca un dicționar ordonat pe litere: căutați `CDROM` și `CD-ROM` și veți vedea care dintre ele va câștiga prin producerea mai multor rezultate. O idee strălucită, dar nu puteți rezolva în mod asemănător controversa dintre `World Wide Web` și `World-Wide Web`, deoarece sistemul tratează și liniile de despărțire și spațiile drept „separatori" și nu poate face diferența dintre ele. Aceia dintre dumneavoastră obișnuiți cu expresiile standard precum cele utilizate de Pearl nu pot nici măcar visa să utilizeze ceva similar în motoarele de căutare.

Pe viitor, motoarele de căutare vor putea oferi opțiuni mai sofisticate, deși interfețele lor de căutare par să se dezvolte în altă direcție, care va fi prezentată în următorul subcapitol.

Prelucrarea interactivă

Mai multe motoare de căutare au dezvoltat proceduri de categorisire a rezultatelor căutării, prin combinarea acestora în grupuri cu „spectrul cuvintelor cheie" similar. Prin selectarea butonului Refine în AltaVista, obțineți o listă cu mai multe categorii în care se încadrează rezultatele dumneavoastră, permițându-vă să specificați dacă doriți să includeți sau să excludeți oricare categorie la următoarea iterație a căutării.

În mod similar, situl Excite (`http://www.excite.com`) vă invită să „selectați cuvintele care să fie adăugate la căutarea dumneavoastră", cu aceste cuvinte cheie suplimentare extrase din rezultatul deja obținut. Această selecție vă permite restrângerea domeniului de căutare mult mai eficient decât o puteți face prin încercarea la întâmplare a diferitelor cuvinte cheie.

Northern Light Search (`http://www.nlsearch.com`) sortează rezultatele căutării și în „dosare", pe baza conținutului acestora și a domeniului URL. Toate aceste caracteristici fac posibilă o căutare puternică prin detectarea interactivă a tendințelor datelor.

Clasificarea

Toate motoarele de căutare clasifică rezultatele astfel încât documentele cele mai semnificative sunt în capul listei. Această sortare este în primul rând bazată pe frecvența cuvintelor cheie din document și în al doilea rând pe distanța aparițiilor cuvintelor cheie față de începutul documentului.

Cu alte cuvinte, dacă un document conține două corespondențe pentru un cuvânt cheie iar altul conține numai o corespondență, primul document va fi mai aproape de capul listei. Dacă două documente au același număr de corespondențe, dar unul are un cuvânt cheie poziționat mai aproape de început (în special în titlul documentului), acesta va fi primul.

În plus față de aceste principii, unele motoare de căutare folosesc factori suplimentari, numiți *acceleratoare de semnificație (relevancy boosters)*, pentru a determina ordinea. De exemplu, HotBot și Infoseek (`http://www.infoseek.com`) favorizează documentele care folosesc eticheta `<META>` față de echivalentele lor care nu o folosesc.

WebCrawler (`http://www.webcrawler.com`) se bazează pe *popularitatea legăturii*. Dacă o pagină este frecvent legată de alte pagini și situri, ea este considerată mai „autorizată" și obține o oarecare prioritate în lista de rezultate. Excite, fiind o combinație între motorul de căutare și director, dă în mod natural întâietate paginilor revăzute în directorul propriu.

În final, toate motoarele de căutare încearcă să combată practicile neprincipiale ale unor Webmasteri care încearcă să păcălească algoritmul de clasificare prin repetarea cuvintelor cheie, pentru a mări frecvența de apariție în documente. Probabil ați observat pagini cu coada de sute de cuvinte cheie repetate (invizibile pentru browser prin schimbarea culorii fontului, dar vizibile totuși pentru motoarele de căutare) sau pagini cu elemente `TITLE` multiple (din nou, numai primul este vizibil pentru browsere, dar toate sunt indexate de păianjen). În prezent, nu numai că asemenea „cuvinte cheie intruse" nu primesc prioritate mare, dar multe motoare de căutare exclud aceste pagini din baza lor de date. (Pentru informații suplimentare privind fenomenul spamming, vedeți Subcapitolul „Controversa Meta", din acest capitol).

Rezultatele

Lista rezultatelor căutării conține titlurile documentelor, adresa URL, rezumatele, uneori data când au fost create documentele (pentru alte motoare de căutare, datele includerii în baza de date), și mărimea documentelor. Pentru compilarea rezumatului documentelor, au fost dezvoltate mai multe soluții.

Multe motoare de căutare folosesc descrierile `META` furnizate de către autorii paginilor (vezi subcapitolul următor pentru mai multe amănunte asupra modului în care se face acest lucru), dar când datele `META` nu sunt disponibile, se iau primele 100 sau 200 de caractere ale textului paginii. Excite se distinge ignorând complet etichetele `<META>`, folosind în schimb algoritmi sofisticați – dar nu prea performanți – care extrag propoziții care par a fi „tema" paginii și le prezintă ca fiind rezumatul paginii.

Soluția care totuși mie mi se pare optimă este cea folosită de Aport, o mașină de căutare rusească (`http://rusia.agama.com/aport/`). În loc să genereze rezumate, Aport listează numai, pentru fiecare document găsit, propozițiile din document care corespund cu interogarea. Într-adevăr, pentru a decide dacă un document merită să fie răsfoit, suntem adesea interesați să vedem care este contextul în care cuvintele cheie sunt regăsite și nu ce fel de document este.

Aport are un număr de alte caracteristici originale față de celelalte motoare de căutare. De exemplu, permite regăsirea unei reconstrucții numai după text a documentului, direct din baza de date a motorului de căutare, în cazul în care documentul original (sau serverul pe care acesta este stocat) este inaccesibil.

Cum se face proiectarea motoarelor de căutare

Dacă ați citit capitolul precedent, „Crearea paginilor Web cu o mare accesibilitate", ați observat probabil asemănarea dintre păianjenii de căutare și persoanele cu handicap: ambele nu au acces decât la conținutul text al paginilor Web. De aceea, cele mai multe recomandări pentru scrierea în limbajul HTML din capitolul referitor la accesibilitate se aplică și la proiectarea prietenoasă pentru căutare.

Furnizarea alternativelor care conțin numai text pentru orice informații din pagina dumneavoastră este o cerință evidentă, deoarece păianjenii scanează numai textul simplu (deși din păcate nu toți indexează textele `alt` asociate imaginilor). Realizarea unui conținut care să fie inteligibil în întregime, folosind numai text, ar putea fi dificilă (este ca și cum ai încerca să convingi pe cineva nu în mod direct ci printr-o scrisoare, fără ajutorul „mediilor" puternice, constituite de gesturi și expresii ale feței), dar vă aduce satisfacție pe termen lung.

Păstrarea cursului normal al textului, fără a-l sacrifica de dragul artificiilor de machetare, este de asemenea foarte important. Acest lucru îmbunătățește șansele ca păianjenii să extragă un rezumat mai bun al documentului dumneavoastră și să facă textul mai potrivit pentru procesarea automată sau categorisire.

Marcarea logică este în mod similar o cerință importantă dacă aveți grijă ca cineva să poată *folosi* documentul dumneavoastră în orice fel, nu numai să-l citească cu un browser grafic. Pe lângă păianjenii celor mai importante motoare de căutare, un mare număr de diferiți roboți și indexori rătăcesc prin Web și mulți dintre ei se bazează pe etichete logice, cum ar fi `H1`, pentru a afla structura datelor dumneavoastră.

Strategii pentru cuvintele cheie

Toate căutările pe Web sunt executate prin cuvinte cheie, așa că cea mai importantă cerință este probabil să se verifice dacă documentele dumneavoastră conțin toate cuvintele cheie care vor fi probabil folosite pentru găsirea documentului. Pentru găsirea cuvintelor cheie pot fi conturate două strategii distincte:

- Prima idee care vine în minte este foarte simplă. Cu cât introduceți mai multe cuvinte cheie în pagină, cu atât este mai bine. Într-adevăr, nu puteți niciodată anticipa ce cuvinte cheie vor veni în mintea utilizatorilor, așa că este o bună idee să vă gândiți întotdeauna la toate posibilitățile privind sinonimele, variantele, termenii și subtermenii generici, precum și conceptele legate de subiectele de bază ale discursului dumneavoastră.

În afară de acest lucru, amintiți-vă că cuvintele cheie pot fi introduse în diferite forme gramaticale, cum ar fi pluralul în locul singularului, pentru substantive. Din motoarele de căutare, numai AltaVista acceptă caracterele de înlocuire pentru a căuta *table* sau *tables*, când se precizează `table*`. Așa că, mai bine rezolvați problema singur, introducând ambele forme în documentul dumneavoastră (această problemă este foarte serioasă pentru celelalte limbi în afara englezei; de exemplu, un verb în limba rusă poate avea 235 de forme. De aceea, cele mai multe motoare de căutare rusești, cum este Aport, menționat anterior, folosesc algoritmi cu inflexiunea cuvintelor în mod implicit, care permit compararea automată a tuturor formelor cuvintelor).

În final, când cuvântul cheie principal este un cuvânt relativ obișnuit (cum ar fi *search*), este probabil ca utilizatorii experimentați în căutare să utilizeze caracteristica expresiei de căutare pentru interogarea cu combinații de cuvinte (cum ar fi *search engines*), mai degrabă

decât cuvinte izolate. De aceea, verificați dacă documentul dumneavoastră conține cele mai obișnuite asociații ale cuvântului cheie principal cu alte substantive, adjective, verbe apropiate.

■ Totuși, cineva se poate gândi la o strategie opusă celei de maximizare a „acoperirii cu cuvinte cheie", tocmai descrisă. Amintiți-vă că unul din factorii clasificării rezultatelor, implementați de motoarele de căutare principale, este *frecvența*, care este calculată ca fiind numărul de apariții ale cuvintelor cheie împărțit la mărimea documentului.

O consecință a acestui calcul este faptul că, dacă două documente conțin același cuvânt cheie (aflat la aceeași distanță de începutul documentului), cel care e mai mic ca dimensiune va avea o importanță mai mare. Acest lucru vă oferă un indiciu: selectați una din paginile rădăcină (introductivă) din situl dumneavoastră și încercați să o faceți cât mai compactă și mai concisă posibil, astfel încât aceasta să prezinte numai esența conținutului, doar cu cuvintele cheie cele mai obișnuite. Această pagină va accelera căutarea acestor cuvinte cheie, astfel încât va atrage mai multe selectări ale întregului sit.

Din acest motiv, cel mai bun lucru pe care-l puteți face este să combinați cele două metode prin stabilirea ambelor tipuri de pagini ale sitului dumneavoastră: cele cu acoperire maximă cu cuvinte cheie și cele cu semnificație maximă în ceea ce privește cuvintele cheie principale.

Aceste două strategii corespund la două tipuri de interogări de căutare, căutările *specifice* și cele *generale*. Unii utilizatori de motoare de căutare caută informații foarte specifice; ei folosesc cuvinte cheie rare, căutări după expresii și caracteristici avansate cum ar fi operatorii booleeni. Aceștia sunt „utilizatorii puternici" cărora ar trebui să le placă paginile dumneavoastră pline de cuvinte cheie.

Alți utilizatori vor să găsească doar o resursă care acoperă un subiect destul de general; ei introduc câteva cuvinte cheie simple, obțin un rezultat avalanșă și răsfoiesc primele legături găsite. Pentru astfel de căutări generale, directoarele Web (cum ar fi Yahoo!) se comportă mai bine decât motoarele de căutare; totuși, mulți utilizatori folosesc motoarele de căutare pentru această operație. Tehnica acceleratoarelor de semnificație, descrisă anterior, poate fi utilă pentru atragerea unor astfel de utilizatori în situl dumneavoastră.

Ați putea fi interesat să vedeți ce cuvinte cheie sunt introduse cel mai frecvent de către utilizatorii motoarelor de căutare pentru a alinia mai bine spectrul cuvintelor cheie la preferințele publicului. Din păcate, această informație (care ar fi interesat mult și din alte puncte de vedere) este considerată strict-secretă de către principalele motoare de căutare – ele nu dezvăluie niciodată lista cu „topul primelor zece cuvinte cheie" de teama (destul de bine întemeiată) față de distribuirea nesolicitată (spamming).

WebCrawler permite numai o privire scurtă asupra fluxului de interogări de căutare în timp real, la intrarea în pagina de căutare (`http://webcrawler.com/WebCrawler/Fun/SearchTicker.html`). Totuși, motoarele de căutare mai puțin importante sunt mai puțin obsedate de confidențialitate, iar unele dintre ele prezintă o statistică a căutării (de exemplu, un motor de căutare rusesc, numit Rambler, prezintă lista sa cu primele 100 de cuvinte de căutare la `http://www.rambler.ru/top.shtml.ru`).

Ultimul sfat în ceea ce privește cuvintele cheie este destul de evident: verificați întotdeauna corectitudinea scrierii. Păianjenii, spre deosebire de cititorii umani, nu pot să „ignore" greșelile de scriere și riscați să ratați o bună parte a audienței, datorită scrierii greșite a unor cuvinte cheie importante. Această sugestie este importantă în mod special în cazurile în care ați adăugat cuvintele dumneavoastră cheie într-o etichetă `<META>`, *după* ce documentul a fost scris, editat și probabil verificat pentru erorile de scriere.

Eticheta <META>

Întorcându-ne la limbajul HTML, s-ar putea să vă întrebați care este sintaxa pentru introducerea cuvintelor cheie. Bineînțeles că textul unei pagini este sursa primară pentru căutare, dar s-ar putea să aveți nevoie să adăugați unele cuvinte cheie, fără a schimba conținutul paginii (schimbarea culorii textului pentru a face cuvintele cheie invizibile în corpul unui document este într-adevăr un artificiu nu prea elegant; vă rog să nu apelați niciodată la el!).

Eticheta `<META>` folosește acestui scop (cât și altor scopuri). *Meta* este cuvântul grecesc pentru „supra", iar eticheta `<META>` este destinată să poarte tot felul de *meta-informații* – adică informații despre (sau „supra") informații. Trebuie să înțelegeți că folosind `<META>` pentru specificarea cuvintelor cheie nu este o convenție HTML, ci numai una din utilizările larg acceptate ale etichetei.

O etichetă `<META>` are următoarea formă:

```
<META name="..." content="...">
```

După cum puteți vedea, numele atributelor etichetei `<META>` sunt cam generice, ceea ce vă permite să folosiți eticheta pentru a exprima aproape orice informație care poate fi reprezentată ca o pereche nume-valoare. De exemplu, puteți folosi etichetele `<META>` pentru a furniza informații despre dumneavoastră (`name="author"`), despre programul pe care-l folosiți pentru a crea fișierul HTML (`name="generator"`) ș.a.m.d.

Iată cum e folosită eticheta `<META>` pentru a prezenta documentul dumneavoastră motoarelor de căutare:

```
<META name="keywords"
  content="searching, search engines, keywords, HTML">
<META name="description"
  content="A description of major web search engines, spiders,
          and search-friendly HTML authoring">
```

Aceste etichete trebuie să fie plasate în elementul `HEAD`. Cuvintele cheie sau frazele din conținutul `content` al etichetei cu atributul `name="keywords"` pot fi separate de virgule pentru o mai bună citire, deși păianjenii ignoră de obicei separatorii. Numărul maxim de cuvinte cheie depinde de motorul de căutare respectiv; pentru unele dintre ele, 25 de cuvinte sau 200 de caractere sunt limita maxim admisă.

Din fericire, cuvintele cheie astfel specificate vor fi adăugate reprezentării ce poate fi căutată a documentului din baza de date a motorului, iar descrierea va fi păstrată ca un rezumat care va fi afișat într-o listă de rezultate (în absența descrierii, cele mai multe motoare de căutare iau prima linie de text de pe pagină).

O altă utilizare a etichetei `<META>` este pentru excluderea unei pagini din atenția păianjenului. Prin adăugarea următoarei etichete, cereți oricărui păianjen care intră în paginile dumneavoastră să treacă peste ea fără să o indexeze:

```
<META name="robots" content="noindex">
```

Totuși, nu toți păianjenii acceptă această convenție. O soluție mult mai fiabilă este să se adauge un fișier `robots.txt` în directorul rădăcină al serverului Web, cu o listă de fișiere care trebuie excluse de la indexare. De exemplu, fișierul dumneavoastră `robots.txt` poate să conțină aceste linii:

```
User-agent: *
Disallow: /dont_index_me.html
Disallow: /hidden_dir/
```

Cu aceste linii nici un robot nu va scana documentul `dont_index_me.html` sau oricare alt document din `/hidden_dir/`. Pentru mai multe informații privind roboții de excludere, mergeți la `http://info.webcrawler.com/mak/projects/robots/exclusion.html`.

Cadrele

Una dintre trăsăturile limbajului HTML merită atenție specială în ceea ce privește accesibilitatea motoarelor de căutare. Aproape jumătate din motoarele de căutare principale nu pot pătrunde în siturile care folosesc cadrele. Pentru acestea, pagina rădăcină a setului de cadre este tot ce se poate vedea și indexa din sit, iar toate paginile de sub rădăcină vor lipsi.

Cea mai bună soluție pentru această problemă (ca și pentru accesibilitatea cadrelor pentru persoanele cu handicap; vezi capitolul 41) este elementul NOFRAMES. Acesta trebuie plasat în cadrul elementului FRAMESET, de obicei înaintea etichetei <FRAME> și poate conține orice text, legături, sau alte materiale. Acest lucru este ceea ce vor vedea motoarele de căutare pe pagină și se va reflecta în baza de date, iar browserele care acceptă cadrele vor ignora conținutul elementului NOFRAMES.

Pentru a face restul conținutului accesibil, ar trebui să furnizați legături la paginile cu cadre din interiorul elementului NOFRAMES. Nu uitați că faceți acest lucru nu numai pentru păianjeni, dar și pentru utilizatorii browserelor ne-grafice, așa că însoțiți legăturile de descrierile corespunzătoare. Unul dintre cadre conține de obicei o bară de navigare cu legături la toate celelalte pagini, așa că în elementul NOFRAMES s-ar putea să fie suficient să-l legați numai la acest document.

Pentru a putea folosi paginile cu cadre în absența contextului cadrelor, nu uitați să le dați propriul element TITLE (acest lucru va îmbunătăți și clasificarea lor în motoarele de căutare).

Controversa Meta

Unul din principalele motoare de căutare, Excite, ignoră orice informație din etichetele <META> și face acest lucru intenționat. Care ar fi raționamentul pentru o astfel de decizie?

Motivul declarat în pagina „Getting Listed" (`http://www.excite.com/Info /listing.html`) este că etichetele <META> pot fi folosite de către cei care fac o distribuție nesolicitată (spammers) pentru a îmbunătăți clasificarea acestora într-un mod necinstit. Pentru Excite, a încerca să faci ca o pagină să apară altfel pentru păianjenii de căutare decât pentru utilizatorii umani este o practică necinstită. Într-adevăr, nimeni nu poate garanta că cuvintele cheie care sunt introduse descriu conținutul documentului dumneavoastră și, în principiu, puteți ușor folosi cuvinte cheie uzuale pentru a mări rata reușitelor, fără nici o îmbunătățire a conținutului paginii.

La prima vedere, această poziție ar părea logică. Dar este într-adevăr? Amintiți-vă că se pot pune ușor oricâte cuvinte cheie „fierbinți" chiar în pagină și dacă nu vreau să atrag atenția cititorilor cu acest mecanism de publicitate, le pot face invizibile prin colorarea acestora în culorile fundalului (așa cum fac deja multe sisteme de spamming, deoarece etichetele <META> nu le permit să introducă prea multe cuvinte cheie). Până la urmă, păianjenii vor indexa întotdeauna ceea ce vreau eu, și interzicerea unei arme nu poate decât să provoace declanșarea cursei înarmării.

Politica motorului Excite se bazează pe presupunerea că fiecare pagină are o „valoare" intrinsecă și că această valoare este evidentă dacă se citește textul din pagină. Dacă acest lucru e adevărat, atunci este normal să se ceară ca păianjenii, pentru a fi capabili să asocieze o valoare justă a „semnificației", să extragă același text ca și oamenii. Deci este acceptat în

mod tacit că un păianjen poate citi, înțelege și evalua textul la fel ca oamenii – și aici se află principala eroare a acestei abordări.

Principalul scop al etichetei <META> este să furnizeze unele informații privind documentul, iar eticheta face acest lucru mai ales pentru calculatoare, care nu pot deduce informația din documentul respectiv. Se presupune că, de exemplu, cuvintele cheie și descrierile prezintă conceptele și subiectele principale din text și nici un program de calculator nu a putut până acum să compileze rezumatul semnificativ sau lista de cuvinte cheie pentru un anume document (în acest context, este interesant de observat că Excite este singurul motor de căutare care utilizează algoritmul inteligenței artificiale pentru compilarea rezumatelor pornind de la textul documentelor).

Într-adevăr, mecanismul <META> este expus abuzurilor, dar până acum este singurul procedeu capabil să ajute calculatoarele să înțeleagă mai bine documentele produse de om. Nu avem altă șansă decât să ne bazăm pe un fel de informație META până când calculatoarele vor ajunge la un nivel de inteligență comparabilă cu a ființelor umane.

Din acest punct de vedere, este interesant să discutăm ultima realizare în domeniul metainformației, Meta Content Framework (MCF). Acest limbaj e folosit pentru descrierea proprietăților metainformațiilor, conexiunilor și relațiilor dintre documente, siturilor, canalelor categoriilor de subiecte și alte informații privind obiectele. MCF a fost realizat de Netscape și a fost trimis la consorțiul W3 în proiect (vezi `http://www.w3.org/TR/NOTE-MCF-XML/`).

MCF poate fi folosit pentru cei care întrețin sistemele de informații închise, cum sunt intraneturile, bazele de date de firmă sau științifice ș.a.m.d. Principala sa promisiune este totuși posibilitatea de a construi o rețea metainformațională pentru întregul Web. Din păcate, fiind dată poziția controversată a actualelor etichete <META> cam primitive din zilele noastre, nu este prea sigur dacă instrumentele sofisticate din MCF vor fi acceptate pe scară largă, chiar dacă vor fi aprobate de W3C.

Rezumat

În acest capitol, am discutat despre motoarele de căutare, un instrument important pentru localizarea și publicitatea informației din Web și procedeele care pot fi folosite pentru a construi pagini HTML mai eficiente și prietenoase la căutare. Au fost prezentate mai ales următoarele subiecte:

- Cum lucrează motoarele de căutare, ce sunt păianjenii de căutare și cum indexează aceștia rețeaua Web
- Ce opțiuni și caracteristici de căutare sunt oferite de principalele motoare de căutare
- Care sunt cele două strategii de selectare și grupare a cuvintelor cheie în paginile dumneavoastră Web
- Cum se folosește eticheta <META> pentru a prezenta cuvintele cheie, rezumatul și alte metainformații corespunzătoare paginii
- De ce problema autorilor care construiesc propriile niveluri de meta-informații este controversată și care sunt implicațiile posibile ale controversei

PARTEA a X-a

Anexe

Anexa A
HTML 4.0 – Referințe

de Lee Anne Phillips, Bob Correll și Will Kelly

În această anexă

- **Funcționalitatea limbajului HTML 670**
- **Browsere și platforme 671**
- **Practica și filozofia limbajului HTML 4.0 671**
- **Elemente structurale (de organizare) 673**
- **Elemente la nivel de bloc 685**
- **Elemente la nivel de text 713**
- **Atribute uzuale și evenimente intrinseci 732**

Limbajul HTML 4.0 este o modernizare a limbajului HTML standard pentru a satisface cerințele de proiectare și dezvoltare ale creatorilor de pagini Web, atât amatori cât și profesioniști. Această anexă conține referințe complete despre elementele și atributele limbajului, așa cum este specificat de către Consorțiul World Wide Web.

OBSERVAȚIE

Această anexă se bazează pe informațiile *din HTML 4.0 Specification W3C Working Draft 17-sep-1997*. Cea mai nouă versiune a acestui document o puteți găsi la adresa: `http://www.w3.org/TR/WD-html40/`.

Pentru a face informația mai ușor accesibilă, în această anexă elementele HTML sunt organizate după funcția lor în codul pe care îl folosiți pentru a crea paginile.

Funcționalitatea limbajului HTML

Există trei tipuri de elemente de bază: structura „organizatorică" a fiecărei pagini, care nu afectează direct interpretarea la nivel înalt a documentului; etichete la nivel de bloc, ce afectează direct machetarea documentului și etichete la nivel de text, care operează inline. În plus, există „entități" mnemonice ce vă permit să accesați diverse caractere folosite uzual în limbile europene și lucrările stiințifice dar nu se regăsesc de obicei pe majoritatea tastaturilor. Aceste entități sunt descrise în Anexa B „Tabelul de referințe pentru compararea browserelor", care conține de asemenea informații despre diferențele dintre browsere. Deoarece limbajul HTML este conceput astfel încât să afișeze pagina într-un browser, înainte de a începe, vom spune câteva cuvinte despre browsere.

Browsere și platforme

Browserele sunt instrumentele care dau putere rețelei Web și, la fel ca alte instrumente, ele sunt de diverse tipuri și dimensiuni pentru a se potrivi pe aproape orice platformă. Indiferent cât plătiți pentru ele, nici unul nu este perfect, nici unul nu va afișa fiecare etichetă exact așa cum credeți că ar trebui afișată, iar unele ar putea avea anumite ciudățenii supărătoare. Așa că înainte de a vă decide pentru un anumit browser, ar trebui să vă uitați la mai multe. Anexa G, „Resurse pentru dezvoltare", conține o mică selecție din care puteți alege. Înainte de a vă hotărî pentru un browser ar trebui să vă gândiți la ce intenționați să-l folosiți.

Dacă aveți un buget redus, încercați unul dintre browserele ieftine ce rulează pe un 386 sau un 486 sub Windows sau pe un echivalent pentru un Mac clasic. Browserele puternic mediatizate, de care vorbește toată lumea, vă vor satisface într-o măsură mai mică pe un PC de genul celor de mai sus dar Opera sau Mozaic va funcționa perfect. De asemenea, dacă doriți să vă uitați la situl vărului de-al doilea din China sau Arabia Saudită, probabil veți încerca Tango sau Internet cu un Accent pentru a putea să vedeți bine glifele.

Dacă doriți să vedeți toate „bunătățile" și „minunățiile" multimedia oferite de limbajul Java, ar trebui să vă decideți pentru o platformă Pentium sau Mac mai nouă, care să conțină fie browserul Netscape Navigator, fie browserul Microsoft Internet Explorer. Suportul pe Web pentru Netscape Navigator e mai bun, dar amândouă vor funcționa bine.

Dacă doriți browsere speciale, cum ar fi browsere audio sau Braille, Anexa G prezintă adresele URL ale unor legături ce v-ar putea ajuta să alegeți cele mai potrivite variante de platformă și software.

Și conținutul acestei cărți vă va ajuta, atât prin etichetele din această anexă, cât și prin numeroasele discuții despre resurse. Browserele vor atinge abia în viitor întreaga capacitate a noilor specificații HTML 4.0 și, de vreme ce sunteți un creator de pagini Web, ar trebui să știți că majoritatea utilizatorilor sunt mulțumiți cu browserul pe care îl au și nu îl vor moderniza niciodată. Mulți creatori de pagini Web au o colecție de browsere pe mai multe platforme, deoarece doresc să-și examineze lucrările în toate variantele în care pot fi afișate.

Practica și filozofia limbajului HTML 4.0

World Wide Web înseamnă comunicare. La fel ca orice alt mediu de conversație, există posibilitatea confuziei sau neînțelegerii. Noile standarde sunt destinate să realizeze o comunicare mai bună și mai cuprinzătoare pentru toți utilizatorii. Pentru un timp, nu a prea contat faptul că limba prestabilită din Web a fost engleza, dar această limitare a restrâns accesul la doar câțiva utilizatori în lume. Dacă studiați etichetele, veți vedea că adăugările au fost alese cu grijă, pentru a permite accesul mai multor utilizatori la conversație și a accepta diferențele dintre modurile în care oamenii comunică și înțeleg. Este într-adevăr impresionant.

OBSERVAȚIE

În următoarea listă, pentru a fi mai ușor, unele etichete sunt trecute împreună cu etichetele înrudite și sunt marcate cu un asterisc acolo unde trebuie să fie sau acolo unde se potrivesc. De asemenea, pentru a fi mai ușor, unele etichete sunt plasate acolo unde trebuie să fie și sunt menționate acolo unde mulți utilizatori le-ar putea căuta.

Unele etichete și atribute nu sunt recomandate ceea ce înseamnă că nu sunt agreate de comitetul de standarde și pe cât posibil ar trebui evitate. Etichetele nerecomandate sunt marcate cu un simbol de grad (°).

În această anexă, etichetele HTML 4.0 sunt grupate astfel:

- Elemente structurale (de organizare)

  Elementele structurale de bază: BODY, HEAD, HTML

  Elementele de titlu: BASE, ISINDEX°, LINK, META, NOSCRIPT, SCRIPT, STYLE, TITLE

  Cadre: FRAMESET, FRAME, IFRAME, NOFRAMES

  Etichete speciale SGML: ! - - (comentariu) - - , ! DOCTYPE

- Elemente la nivel de bloc

  Elemente de bază la nivel de bloc : ADDRESS, BLOCKQUOTE, CENTER°, DIV, FIELDSET, H1-H6, HR, ISINDEX°*, P, PRE

  Liste : DD, DIR°, DL, DT, LI, MENU°, OL, UL

  Formulare : BUTTON, FIELDSET, FORM, INPUT, LABEL, LEGEND, OPTION, SELECT, TEXTAREA

  Tabele: CAPTION, COL, COLGROUP, TABLE, TBODY, TD, TFOOT, TH, THEAD, TR

- Elemente la nivel de text

  Marcatoare de text: B, BIG, I, S°, SMALL, STRIKE°, TT, U°

  Marcatoare de expresie: ACRONYM, ADDRESS*, BLOCKQUOTE*, CITE, CODE, DEL, DFN, EM, INS, KBD, PRE*, SAMP, STRONG, VAR

  Marcatoare speciale: A, APPLET°, BASEFONT°, BDO, BR, FONT°, IFRAME*, IMG, NOSCRIPT*, OBJECT, Q, SCRIPT*, SPAN, SUB, SUP

  Hărți imagine pe partea clientului: AREA, MAP

  Text pentru controlul formularelor: BUTTON*, INPUT*, LABEL*, SELECT*, TEXTAREA*

În fiecare secțiune, elementele sunt listate în ordine alfabetică și sunt furnizate următoarele informații:

- Eticheta propriu-zisă și eticheta sa de încheiere, dacă are una. Dacă o etichetă este opțională atunci ea este trecută între paranteze drepte, dar parantezele nu fac parte din nici o etichetă.
- Utilizare: Reprezintă o descriere generală a elementului.
- Sintaxa: Furnizează un scurt exemplu despre cum se scrie eticheta în pagină. Elementele opționale sunt trecute între paranteze drepte. Atributele opționale sunt scrise cu caractere italice – *atribute* – pentru a arăta că atributele actuale, listate ulterior în acea secțiune, apar în acel loc.
- Eticheta de start/stop: Precizează dacă aceste etichete sunt cerute, opționale sau interzise.
- Atribute: Prezintă atributele elementului cu o scurtă descriere a efectelor lor. Dacă există o valoare prestabilită, ea este scrisă **cu aldine.**
- Intrinseci: Prezintă evenimentele intrinseci care pot fi detectate.

OBSERVAȚIE

Limbajul HTML 4.0 introduce câteva funcții și evenimente intrinseci noi care se aplică unui număr mare de elemente. Ele sunt apelate prin nume la sfârșitul listei specifice, dar, deoarece au aproape întotdeauna exact aceeași descriere și același efect, pentru o explicație completă consultați ultima secțiune a acestei anexe, „Atribute uzuale și evenimente intrinseci".

- Gol: Arată că elementul trebuie să fie gol.

- Observații: Introduce anumite observații atunci când se folosește elementul și indică dacă elementul este nou sau nerecomandat.
- Introdus: Versiunea limbajului HTML care a standardizat elementul respectiv.
- Browsere: Indică ce versiune a browserelor uzuale (Netscape Navigator sau Microsoft Internet Explorer) acceptă elementul din limbajul HTML.
- Exemplu: Prezintă un scurt exemplu de cod care arată cum este folosită eticheta.

OBSERVAȚIE

Elementele care sunt evidențiate sunt scrise **cu aldine.** Pentru o mai bună ilustrare, elementele și atributele nerecomandate sunt uneori folosite în exemplu, dar sunt scrise cu litere *italice*, pentru a arăta că nu mai sunt recomandate.

Elemente structurale (de organizare)

Limbajul HTML se bazează pe câteva elemente pentru a da o anumită structură unui document (spre deosebire de metoda de structurare a textului în interiorul documentului) și furnizează informația care este folosită apoi de browser sau de motoarele de căutare.

Elementele structurale de bază: BODY, HEAD, HTML

Acestea sunt etichetele de bază după care pot fi găsite toate paginile. Ele sunt toate unice deoarece pot fi aproape întotdeauna deduse din context și multe pagini vor funcționa foarte bine fără oricare dintre ele. Totuși ar fi foarte bine să fie puse în pagini, deoarece ele formează un anumit șablon de gândire și previn apariția greșelilor ce ar putea face ca anumite browsere să afișeze pagina eronat.

[<BODY>]...[</BODY>]

Utilizare:	Conține corpul documentului
Sintaxa :	<BODY [*atribute*]>*corpul documentului HTML*</BODY>
Eticheta de start/stop:	Opțională/Opțională
Atribute :	background="..." Nerecomandat. Este adresa URL pentru imaginea din fundal.
	bgcolor="..." Nerecomandat. Stabilește culoarea fundalului.
	text="..." Nerecomandat. Culoarea textului.
	link="..." Nerecomandat. Culoarea legăturii.
	vlink="..." Nerecomandat. Culoarea legăturii vizitate.
	alink="..." Nerecomandat. Culoarea legăturii active.
	class=, id=, style=, title=, lang=, dir= Atribute uzuale.
Intrinseci:	onload="..." Reprezintă un eveniment intrinsec.
	onunload="..." Reprezintă un eveniment intrinsec activat când documentul este descărcat.
	onclick=, ondblclick=, onmousedown=, onmouseup=, onmouseover, onmousemove=, onmouseout=, onkeypress=, onkeydown=, onkeyup= Evenimente intrinseci uzuale.

Gol: Nu

Observații: Poate fi definit un singur element BODY iar acesta trebuie să fie plasat după elementul HEAD (titlu). Elementul BODY (corp) poate fi înlocuit cu un element FRAMESET (configurare de cadru). Se recomandă folosirea foilor de stil și nu a atributelor de prezentare.

Introdus: HTML 1.0

Browsere: Netscape Navigator 0.9, 1.x, 2.x, 3.x, 4.x; Microsoft Internet Explorer 1.x, 2.x, 3.x, 4.x.

Exemplu:

```
<BODY>
  <P align=center>
    In Scarlet town, where I was born<BR>
    There was a fair maid dwellin',<BR>
    Made every youth cry <I>Well-a-way!</I><BR>
    Her name was Barbara Allen.
</BODY>
```

[<HEAD>]...[</HEAD>]

Utilizare: Acesta este titlul documentului. El conține alte elemente care dau informații utilizatorilor și motoarelor de căutare.

Sintaxa: `[<HEAD [atribute]>] informatia din titlu [</HEAD>]`

Eticheta de start/stop: Opțională/Opțională

Atribute: `profile="..."` Este adresa URL care precizează locația datelor META.

`lang=, dir=` Atribute uzuale.

Gol: Nu

Observații: Poate fi definit un singur element HEAD (titlu) pentru un document. Acesta trebuie să fie plasat după eticheta HTML de deschidere și să preceadă BODY.

Introdus: HTML 1.0

Browsere: Netscape Navigator 0.9, 1.x, 2.x, 3.x, 4.x; Microsoft Internet Explorer 2.x, 3.x, 4.x

Exemplu:

```
<HTML>
<HEAD>
  <TITLE>Acesta este textul care va aparea pe bara de titlu a
  ➥browserului </TITLE>
</HEAD>
<BODY>
  <P>Elementul titlu contine textul titlului care apare în
     linia pentru subiect a browserului.
  <P>Trebuie sa fie exact, descriptiv si sa dea cititorilor
     o referinta despre ce contine pagina. Majoritatea
     browserelor plaseaza titlul în semnul de carte, în caz
     ca utilizatorul doreste sa salveze situl Dvs. si sa
     revina mai târziu. Puteti sa-l ajutati prin crearea unui
     titlu atragator si distinctiv care identifica situl
     si spune în cateva cuvinte cine sunteti.
  </BODY>
</HTML>
```


[<HTML>]...[</HTML>]

Utilizare: Elementul HTML conține întregul document.

Sintaxa: `[<HTML [atribute]>]corpul documentului si alte elemente[</HTML>]`

Etichete de start/stop: Opțională/Opțională

Atribute: `version="..."` Este adresa URL pentru definirea tipului documentului, precizând versiunea limbajului HTML folosită pentru a crea documentul.

`lang=, dir=` Atribute uzuale.

Gol: Nu

Observații: Informația despre versiune este conținută și în declarația `<!DOCTYPE...>`, așa că nu este esențială.

Introdus: HTML 1.0

Browsere: Netscape Navigator 0.9, 1.x, 2.x, 3.x, 4.x; Microsoft Internet Explorer 2.x, 3.x, 4.x.

Exemplu:

```
<HTML>
  <HEAD>
    <TITLE>Galeria</TITLE>
  </HEAD>
  <BODY bgcolor="#FFFFFF">
    <H1 align="center"> Galeria</H1>
  </BODY>
</HTML>
```

Elementele de titlu: BASE, ISINDEX, LINK, META, NOSCRIPT, SCRIPT, STYLE, TITLE

Elementele de titlu se află în antetul unui document HTML și au un efect redus, dacă acesta există, la formatarea directă a documentului. Totuși, în cazul elementelor SCRIPT și STYLE, ele pot modifica dinamic formatarea implicită.

<BASE>

Utilizare: Toate celelalte adrese URL din document sunt alcătuite relativ la această locație.

Sintaxa: `<BASE [atribute]>`

Eticheta de start/stop: Cerută/Interzisă

Atribute: `href="..."` Reprezintă adresa URL a resursei legate.

`target="..."` Determină unde va fi afișată resursa.
(nume dat de utilizator,`_blank`,`_parent`,**`_self`**,`_top`).

Gol: Da, eticheta de încheiere este interzisă.

Observații: Se află în titlul documentului.

Introdus: HTML 2.0

Browsere: 2.0, 3.0, 4.0. Netscape Navigator 1.0, 2.x, 3.x, 4.x; Microsoft Internet Explorer

Exemplu:

```
<HTML>
  <HEAD>
    <TITLE> Base Tag Anatomy</TITLE>
    <BASE href ="http://www.example.com/anatomy/">
</HEAD>
...
```

<ISINDEX> [0]

Utilizare: Creează un formular simplu care solicită utilizatorului o intrare.

Sintaxa: `<ISINDEX [atribute]>`

Eticheta de start/stop: Cerută/Interzisă

Atribute: `prompt="..."` Nerecomandat. Furnizează un șir de tip prompt pentru câmpul de intrare.

`class=,id=,style=,title=,lang=,dir=` Atribute uzuale.

Gol: Da, eticheta de încheiere este interzisă.

Observații: Nerecomandat. Folosiți în loc elementul `INPUT` (intrare) într-un formular (`FORM`).

Introdus: HTML 3.2

Browsere: Netscape Navigator 1.0, 2.x, 3.x, 4.x; Microsoft Internet Explorer 2.0, 3.0, 4.0.

Exemplu:

```
<HTML>
  <HEAD>
    <TITLE>Index de cautare</TITLE>
    <ISINDEX prompt="Introduceti sirul de cautare:">
  </HEAD>
    <BODY>
  ...
  </BODY>
</HTML>
```

<LINK>

Utilizare: Definește relația dintre o legătură și o resursă.

Sintaxa: `<LINK [atribute]>`

Eticheta de start/stop: Cerută/Interzisă

Atribute:

`href="..."` Adresa URL a resursei.

`rel="..."` Reprezintă tipurile de legături înainte.

`rev="..."` Reprezintă tipurile de legături înapoi.

`type="..."` Definește tipul conținutului Internet.

`media="..."` Definește mediul destinației (`screen` (ecran), `print` (imprimantă), `projection` (proiecție), `Braille`, `speech` (vorbit), `all` (toate)).

`target="..."` Determină unde va fi afișată resursa.
(nume definit de utilizator,`_blank`,`_parent`,**`_self`**,`_top`).

`class=,id=,style=,title=,lang=,dir=` Atribute uzuale.

Intrinseci: `onclick=,ondblclick=,onmousedown=,onmouseup=,onmouseover=,onmousemove=,onmouseout=,onkeypress=,onkeydown=,onkeyup=` Evenimente intrinseci uzuale.

Tipurile de legături:

`contents` Legătura se referă la un document structurat ca un cuprins.

`index` Legătura se referă la un document structurat ca un index pentru documentul curent.

`glossary` Legătura se referă la un document structurat ca un glosar pentru documentul curent.

`copyright` Legătura se referă la o declarație de copyright pentru documentul curent.

`next` Legătura se referă la un document care este următorul dintr-o serie ordonată de documente.

`previous` Legătura se referă la documentul anterior dintr-un set ordonat de documente.

`start` Legătura se referă la primul document dintr-un set ordonat de documente.

`help` Legătura se referă la documentul care conține informațiile de tip Help.

`bookmark` Legătura se referă la un semn de carte inclus în document.

Gol: Da, eticheta de încheiere este interzisă.

Observații: Această etichetă este localizată în titlul documentului. Teoretic, puteți crea sisteme de navigare sofisticate cu aceste atribute, inclusiv selectarea automată a limbii și alte facilități. Din nefericire, majoritatea browserelor nu prea utilizează informația.

Introdus: HTML 2.0

Browsere: 2.0, 3.0, 4.0. Netscape Navigator 1.0, 2.x,3.x,4.x; Microsoft Internet Explorer

Exemplu:

```
...
<HEAD>
  <LINK href="contents.html" rel="next" rev="prev">
```

```
</HEAD>
...>
```

<META>

Utilizare: Furnizează informații despre document.

Sintaxa: `<META [atribute]>`

Eticheta de start/stop: Cerută/Interzisă

Atribute:

`http-equiv="..."` Reprezintă numele titlului de la adresa HTTP care răspunde.

`name="..."` Reprezintă numele informației meta.

`content="..."` Reprezintă conținutul informației meta.

`scheme="..."` Atribuie o schemă pentru a interpreta datele `meta`.

`lang=, dir=` Atribute uzuale.

Gol: Da, eticheta de încheiere este interzisă.

Observații: Conținutul elementului `<META>` este folosit în scopuri de indexare de către mulți păianjeni și motoare de căutare pe Web.

Introdus: HTML 2.0

Browsere: Netscape Navigator 1.0, 2.x, 3.x, 4.x; Microsoft Internet Explorer 2.0, 3.0, 4.0.

Exemplu:

```
<HTML>
  <HEAD>
    <TITLE> Author, Author</title>
    <META name="auhor" content="Joe Sample">
    <META name="keywords" content="research writing html">
  </HEAD>
  <BODY>
<H1>Example Page</H1>
<H2>by Joe Sample</H2>
    ...
  </BODY>
</HTML>
```

<NOSCRIPT>...</NOSCRIPT>

Utilizare: Elementul `NOSCRIPT` furnizează un conținut alternativ pentru browserele care nu pot rula un script.

Sintaxa: `<NOSCRIPT>continutul alternativ</NOSCRIPT>`

Eticheta de start/stop: Cerută/Cerută

Atribute: Nici unul

Gol: Nu

Observații: Acesta este într-adevăr un element text inline, plasat aici pentru ca ambele etichete `<SCRIPT>` să fie împreună.

Introdus: HTML 4.0

Browsere: Netscape Navigator 3.0, 4.0; Microsoft Internet Explorer 4.0.

Exemplu:

```
<HTML>
  <HEAD>
    <TITLE> Noscript Demonstration</TITLE>
  </HEAD>
  <BODY>
    <SCRIPT type ="text/javascript">
     <!--
      document.write ("<B>Hello, world.</B>")
     /-->
    </SCRIPT>
    <NOSCRIPT>
      <P>Hello, world.
    </NOSCRIPT>
  </BODY>
</HTML>
```

<SCRIPT>...</SCRIPT>

Utilizare: Elementul `SCRIPT` conține secvențele de comenzi pe partea clientului, care sunt executate de către browser .

Sintaxa: `<SCRIPT [atribute]>comenzile script</SCRIPT>`

Eticheta de start/stop: Cerută/Cerută

Atribute:

`type="..."` Tipul de conținut Internet al limbajului folosit în script.

`language="..."` Nerecomandat. Limbajul folosit în script, nerecomandat în favoarea atributului `type`.

`src="..."` Este adresa URL pentru scriptul extern.

Gol: Nu

Observații: Puteți stabili limbajul prestabilit folosit în script, în elementul `META` .

Introdus: HTML 4.0

Browsere: Netscape Navigator 1.0, 2.0, 3.x, 4.0; Microsoft Internet Explorer 3.x, 4.0

Exemplu:

```
...
<SCRIPT type ="text/javascript">
 <!--
 document.write ("<B>Hello, world.</B>")
 /-->
</SCRIPT>
...
```

<STYLE>...</STYLE>

Utilizare: Creează o foaie de stil internă.

Sintaxa: `<STYLE [atribute]>continutul foii de stil</STYLE>`

Eticheta de start/stop: Cerută/Cerută

Atribute:
- `type="..."` Reprezintă tipul de conținut Internet.
- `media="..."` Definește mediul de destinație (`screen`, `print`, `projection`, `Braille`, `speech`, `all`.
- `title="..."` Reprezintă titlul stilului.
- `lang=`, `dir=` Atribute uzuale.

Gol: Nu

Observații: Este localizat în elementul HEAD. Este o idee bună să adăugați foii de stil un comentariu, deși acesta nu este strict necesar, din cauză că unele browsere, la vederea unui text în titlu, presupun că s-a ajuns la corpul paginii și încep interpretarea.

Introdus: HTML 4.0

Browsere: Netscape Navigator 4.0; Microsoft Internet Explorer 3.x, 4.0.

Exemplu:

```
<HTML>
  <HEAD>
    <TITLE>Intro to Style Sheets</title>
    <STYLE type="text\css">
      <! --
        /* make all paragraphs red */
        p:(color:red:font-style:bold)
        -->
    </STYLE>
  </HEAD>
</HTML> occur
```

<TITLE>...</TITLE>

Utilizare: Acesta este titlul pe care îl veți da paginii Web. Elementul TITLE, aflat în elementul HEAD, este afișat în bara de titluri a ferestrei browserului.

Sintaxa: `<TITLE [atribute]>titlul paginii</TITLE>`

Eticheta de start/stop: Cerută/Cerută

Atribute: `lang`, `dir`

Gol: Nu

Observații: Această etichetă este cerută la fiecare document. Este permisă definirea doar a unui singur titlu pentru un document.

Introdus: HTML 4.0

Browsere: Netscape Navigator 1.0, 2.x, 3.x, 4.x; Microsoft Internet Explorer 2.0, 3.0, 4.0.

Exemplu:

```
...
<HEAD>
  <TITLE>Titlul apare în bara de titlu a browserului</TITLE >
</HEAD>
...
```

Cadre: FRAMESET, FRAME, IFRAME, NOFRAMES

Cadrele creează „panouri" separate în fereastra browserului și sunt folosite pentru a afișa conținutul documentelor din diverse surse.

<FRAME>

Utilizare: Definește un cadru.

Sintaxa: `<FRAME [atribute]>`

Eticheta de start/stop: Cerută/Interzisă

Atribute:
- `name="..."` Reprezintă numele unui cadru.
- `src="..."` Reprezintă sursa ce va fi afișată în cadru.
- `frameborder="..."` Arată dacă se afișează sau nu bordura dintre cadre (0, 1).
- `marginwidth="..."` Stabilește spațiul dintre bordura cadrului și conținut.
- `marginheight="..."` Stabilește spațiul dintre bordura cadrului și conținut.
- `noresize` Dezactivează dimensionarea.
- `scrolling="..."` Stabilește prezența barei de derulare (`auto`, `yes`, `no`).

Gol: Da, eticheta de încheiere este interzisă.

Observații: Cadrele au fost remarcate prima oară ca fiind una dintre cele mai populare extensii create de Netscape, lansate împreună cu Netscape Navigator 2.0. Ulterior, Corporația Netscape Communications le-a propus consorțiului W3C pentru a fi integrate în următorul standard al limbajului HTML.

Introdus: HTML 4.0

Browsere: Netscape Navigator 2.0, 3.x, 4.x; Microsoft Internet Explorer 3.0, 4.0.

Exemplu:

```
<!DOCTYPE html public"-//IETF//DTD  HTML//EN">
<HTML>
  <HEAD>
    <TITLE>Technical Writing</TITLE>
    <META name="FORMATTER" content="Microsoft FrontPage 2.0">
  </HEAD>
  <FRAMESET rows ="12%, * , 12%">
    <FRAME src="frtop.htm"  name="top" noresize>
    <FRAMESET cols="35% , 65%">
      <FRAME src="frconten.htm" name="contents">
      <FRAME src="frmain.htm" name="main">
```

```
    </FRAMESET>
    <FRAME src="frbottom.htm" name="bottom" noresize>
  <NOFRAMES>
    <BODY>
      <P> Aceasta pagina Web contine cadre dar browserul
      dumneavoastra nu accepta cadre. Va rog sau mergeti la
      versiunea fara cadre a paginii <A ref ="tw _nonf.htm">
      Technical Writing Resources</A>.
    </BODY>
  </NOFRAMES>
  </FRAMESET>
</HTML>
```

<FRAMESET>...</FRAMESET>

Utilizare: Definește dispunerea cadrelor într-o fereastră.

Sintaxa: `<FRAMESET [atribute]>continutul definirii cadrului</FRAMESET>`

Eticheta de start/stop: Cerută/Cerută

Atribute:

`rows="..."` Reprezintă numărul de rânduri.

`cols="..."` Reprezintă numărul de coloane.

`onload="..."` Reprezintă evenimentul intrinsec activat atunci când se încarcă documentul.

`onunload="..."` Reprezintă evenimentul intrinsec activat atunci când se descarcă documentul.

Gol: Nu

Observații: Elementele FRAMESET pot fi imbricate.

Introdus: HTML 4.0

Browsere: Netscape Navigator 2.0, 3.x, 4.x; Microsoft Internet Explorer 3.0, 4.0.

Exemplu:

```
<HTML>
  <HEAD>
    <TITLE>Cele trei varste</TITLE>
  </HEAD>
  <FRAMESET rows="*, *, *">
    <FRAME src="copilarie.htm">
    <FRAME src="maturitate.htm">
    <FRAME src="batranete.htm">
  </FRAMESET>
  <NOFRAMES>
    <BODY>
      <P>
      Aceasta este versiunea fara cadre a paginii"Trei
      varste"<BR>
      Alegeti va rog una din urmatoarele destinatii:<BR>
      <A href="copilarie.htm">Copilarie</A><BR>
      <A href="maturitate.htm">Maturitate</A><BR>
      <A href="batranete.htm">Batranete</A><BR>
    </BODY>
  </NOFRAMES>
</HTML>
```


<IFRAME>...</IFRAME>

Utilizare: Creează un cadru inline.

Sintaxa: `<IFRAME [atribute]>continutul cadrului inline</IFRAME>`

Eticheta de start/stop: Cerută/Cerută

Atribute:

`name="..."` Reprezintă numele cadrului.

`src="..."` Reprezintă sursa care va fi afișată într-un cadru.

`frame border="..."` Afișează sau nu bordura dintre cadre (0, 1).

`marginwidth="..."` Stabilește spațiul dintre bordura cadrului și conținut.

`marginheight="..."` Stabilește spațiul dintre bordura cadrului și conținut.

`scrolling="..."` Determină prezența barei de derulare (`auto`, `yes`, `no`).

`align="..."` Nerecomandat. Controlează alinierea (`left` (stânga), `center` (centru), `right` (dreapta), `justify` (aliniere la stânga și la dreapta)).

`height="..."` Reprezintă înălțimea.

`width="..."` Reprezintă lățimea.

Gol: Nu

Observații: Elementul IFRAME este într-adevăr un element inline, dar este util să fie plasat și aici.

Introdus: HTML 4.0

Browsere: Microsoft Internet Explorer 3.0, 4.0.

Exemplu:

```
<HTML>
  <HEAD>
    <TITLE>Trei varste</TITLE>
  </HEAD
  <BODY>
    <P> Aceste trei varste, copilarie
      <IFRAME src="copilarie.htm">
       maturitate,
      <IFRAME src="maturitate.htm">
       si batranete
      <IFRAME src="batranete.htm">.
  </BODY>
</HTML>
```

<NOFRAMES>...</NOFRAMES>

Utilizare: Asigură un conținut alternativ atunci când cadrele nu sunt acceptate.

Sintaxa: `<NOFRAMES>continutul afisat in browsere care nu accepta cadre </NOFRAMES>`

Eticheta de start/stop: Cerută/Cerută

Atribute: Nici unul

Gol: Nu

Introdus: HTML 4.0

Browsere: Netscape Navigator 2.x, 3.x, 4.x; Microsoft Internet Explorer 3.0, 4.0.

Exemplu:

```
...
<NOFRAMES>
  <BODY>
    <P> Aceasta pagina Web contine cadre, dar browserul
    dumneavoastra nu accepta cadre. Va rog sa mergeti la
    versiunea fara cadre a paginii <A ref ="tw _nonf.htm">
    Technical Writing Resources </A>.
  </BODY>
</NOFRAMES>
...
```

Etichete speciale SGML : !-- (*comentariu*) --, !DOCTYPE

Aceste etichete sunt amândouă cazuri speciale : limbajul SGML (Standard Generalized Markup Language), face parte din limbajul HTML. Comentariile SGML, în particular, sunt o sursă de comportare neobișnuită în browsere, deoarece mulți programatori nu prea cunosc modul în care acestea funcționează. Cel mai bine ar fi să evitați folosirea etichetei !DOCTYPE, deoarece nici unul dintre browserele principale nu o acceptă și ar putea face ca paginile dumneavoastră să nu treacă de verificarea validatorilor HTML locali sau online care acceptă această etichetă.

<!-- ... -->

Utilizare: Este un comentariu SGML. Este folosit pentru a insera observații sau scripturi care nu sunt afișate de browser.

Sintaxa: `<![-.....-] *>`

Eticheta de start/stop: Nu este cazul

Atribute: Nici unul

Gol: Nu este cazul

Observații : Comentariile sunt un caz special. Etichetele pentru comentarii sunt de fapt două cratime (- -) care încadrează fiecare comentariu, dar comentariul trebuie să fie inclus într-o comanda SGML : <!...>. Primele două cratime trebuie să urmeze după <! fără spații libere și trebuie să urmeze un spațiu liber imediat după ele. Cratimele pentru comentariu trebuie puse în perechi, astfel că atunci când

există mai multe comentarii, vor fi patru cratime (- - - -) între fiecare. Multe browsere nu tratează comentariile corect; cele mai obișnuite erori, constau în nepermiterea comentariilor multilinie, părăsirea comentariului la al doilea set de cratime sau părăsirea comentariului la primul >. Toate aceste comportări reprezintă erori, dar trebuie să fie luate în considerare de către programator.

Introdus: HTML 2.0

Browsere: Netscape Navigator 1.0, 2.x, 3.x, 4.x; Microsoft Internet Explorer 1.x, 2.x, 3.x, 4.x.

Exemplu:

```
<HTML>
<!-- Se pare ca nimeni nu mai citeste comentariile -->
<!-- Acest lucru este valabil si pentru comentariile
     multi-linie cum e acel pe care îl cititi chiar acum.-->
<!-- Aceasta este o serie de comentarii --
     -- desi multe browsere --
     -- nu le interpreteaza corect -->
<!-- Aceasta este o eticheta <P> într-un comentariu, pe care
     multe browsere nu o trateaza  corect.-->
</HTML>
```

<!DOCTYPE...>

Utilizare: Informația referitoare la versiune apare pe prima linie a unui document HTML și este mai degrabă o declarație SGML decât un element.

Sintaxa: `<!DOCTYPE html public "[numele DTD standard]">`

Eticheta de start/stop: Cerută/Nici una

Atribute: Nici unul

Gol: Nu este cazul

Observații: Această etichetă poate fi omisă deoarece, în prezent, doar câteva browsere o acceptă. De fapt, poate cauza probleme dacă nu aveți motive pentru a o valida față de un anumit DTD, deoarece nu există o modalitate de a spune că încercați să acoperiți toate bazele prin includerea elementelor din mai multe baze.

Introdus: HTML 1.0

Browsere: Netscape Navigator 0.9, 1.x, 2.x, 3.x, 4.x; Microsoft Internet Explorer 2.x, 3.x, 4.x.

Exemplu:

```
<!DOCTYPE html public"-//IETF//DTD HTML 4.0//EN">
<HTML>
...
```

Elemente la nivel de bloc

Elementele la nivel de bloc determină o întrerupere în fluxul textului și afectează formatarea documentului.

Elemente bloc de bază : ADDRESS, BLOCKQUOTE, CENTER°, DIV, FIELDSET, H1 - H6, HR, ISINDEX°*, NOSCRIPT*, P, PRE

Elementele bloc de bază modifică formatarea documentului și determină o întrerupere în fluxul textului dar nu sunt parte integrantă din alte elemente bloc sau nu conțin alte elemente bloc.

<ADDRESS>...</ADDRESS>

Utilizare: Asigură un format special pentru autor sau pentru informațiile de contact sau permite extragerea automată a unei adrese printr-un proces.

Sintaxa: `<ADDRESS [atribute]>adresa sau informatia de contact </ADDRESS>`

Eticheta de start/stop: Cerută/Cerută

Atribute: `class=, id=, style=, title=, lang=, dir=` Atribute uzuale.

Intrinseci: `onclick=, ondblclick=, onmousedown=, onmouseup=, onmouseover=, onmousemove=, onmouseout=, onkeypress=, onkeydown=, onkeyup=` Evenimente intrinseci uzuale.

Gol: Nu

Observații: Acest element produce o întrerupere în fluxul textului, așa că este introdus aici. Elementul `BR` este folosit în mod uzual în interiorul elementului `ADDRESS`, pentru a întrerupe liniile unei adrese. Unele browsere redau conținutul ca un text scris cu caractere italice.

Introdus: HTML 2.0

Browsere: Netscape Navigator 1.0, 2.0, 3.x, 4.x; Microsoft Internet Explorer 2.0, 3.0, 4.0.

Exemplu:

```
<ADDRESS>
  author @ example.com <BR>
  123 Main Street <BR>
  Anytown, IN <BR>
  USA
</ADDRESS>
```

<BLOCKQUOTE>...</BLOCKQUOTE>

Utilizare: Folosit pentru a afișa citatele lungi.

Sintaxa: `<BLOCKQUOTE [atribute]>textul citatului</BLOCKQUOTE>`

Eticheta de start/stop: Cerută/Cerută

Atribute: `cite= "..."` Adresa URL a textului citat sau explicația modificării textului.

`class=, id=, style=, title=, lang=, dir=` Atribute uzuale.

Intrinseci: `onclick=, ondblclick=, onmousedown=, onmouseup=, onmouseover=, onmousemove=, onmouseout=, onkeypress=, onkeydown=, onkeyup=` Evenimente intrinseci uzuale.

Gol: Nu

Observații: Este redat în mod obișnuit, la o distanță de ambele margini. Este înrudit dar nu este identic cu eticheta `<Q>`, care marchează un citat inline.

Introdus: HTML 2.0.

Browsere: Netscape Navigator 1.0, 2.0, 3.x, 4.x; Microsoft Internet Explorer 2.0, 3.0, 4.0.

Exemplu:

```
...
<BLOCKQUOTE  cite ="http : www.teresaofavila.com/">
  Imi doresc sa pot scrie cu ambele maini, ca astfel sa nu
  ➥uit un lucru în timp ce spun altul. &mdash;
  ➥Teresa of Avila.
</BLOCKQUOTE>
```

<CENTER>...</CENTER>°

Utilizare: Folosit pentru a centra un element HTML.

Sintaxa: `<CENTER [atribute]>textul sau elementele centrate</CENTER>`

Eticheta de start/stop: Cerută/Cerută

Atribute: `class=, id=, style=, title=, lang=, dir=` Atribute uzuale.

Intrinseci: `onclick=, ondblclick=, onmousedown=, onmouseup=, onmouseover=, onmousemove=, onmouseout=, onkeypress=, onkeydown=, onkeyup=` Evenimente intrinseci uzuale.

Gol: Nu

Observații: Nerecomandat. Este definit în HTML 4.0 ca un sinonim pentru elemen `<DIV align ="center">`, care ar trebui să fie folosit în locul lui.

Introdus: HTML 2.0.

Browsere: Netscape Navigator 1.1, 2.0, 3.x, 4.x; Microsoft Internet Explorer 2.0, 3.0, 4.0.

Exemplu:

```
<CENTER>
  <A href ="http ://www.example.org/pancake/">Pancake
Breakfast!<A>
</CENTER>
```

<DIV>...</DIV>

Utilizare: Elementul `DIVISION` este folosit pentru a adăuga o structură unei secțiuni de text.

Sintaxa: `<DIV [atribute]>textul structurat</DIV>`

Eticheta de start/stop: Cerută/Cerută

Atribute: `align= "..."` Este nerecomandat. Controlează alinierea (`left, center, right, justify`).

`class=, id=, style=, title=, lang=, dir=` Atribute uzuale.

Intrinseci: `onclick=, ondblclick=, onmousedown=, onmouseup=, onmouseover=, onmousemove=, onmouseout=, onkeypress=, onkeydown=, onkeyup=` Evenimente intrinseci uzuale.

Gol: Nu

Observații: Nu poate fi folosit într-un element P. Atributul `align` este nerecomandat pentru a putea controla alinierea, în favoarea foilor de stil.

Introdus: HTML 2.0

Browsere: Netscape Navigator 2.0, 3.x, 4.x; Microsoft Internet Explorer 2.x, 3.x, 4.x.

Exemplu:

```
<HTML>
  <DIV align="left">
    <P> ... for as ye think I fear too much,
  </DIV>
  <DIV align ="right">
    <P> be you well aware that you fear not as for too little.
  </DIV>
  <DIV align="center">
    <P>-Elizabeth Grey
  </DIV>
</HTML>
```

<H1>...</H1> până la <H6>...</H6>

Utilizare: Cele șase anteturi (H1 este cel mai important) sunt folosite în corp (`BODY`) pentru a structura ierarhic informația.

Sintaxa: `<H1 [atribute]>textul de antet</H1>`

Eticheta de start/stop: Cerută/Cerută

Atribute: `align = "..."` Este nerecomandat. Controlează alinierea (`left, center, right, justify`).

`class=, id=, style=, title=, lang=, dir=` Atribute uzuale.

Intrinseci: `onclick=, ondblclick=, onmousedown=, onmouseup=, onmouseover=, onmousemove=, onmouseout=, onkeypress=, onkeydown=, onkeyup=` Evenimente intrinseci uzuale.

Gol: Nu

Observații: Browserele vizuale afișează mărimea anteturilor în funcție de importanța lor, cu `H1` fiind cel mai important, iar `H6` cel mai puțin important. Pentru compatibilitate, anteturile trebuie să înceapă de la `H1` și să scadă secvențial până la cea mai mică valoare. Atributul `align` este nerecomandat în favoarea controlului alinierii folosind foile de stil.

Introdus: HTML 1.0

Browsere: Netscape Navigator 1.0, 2.x, 3.x, 4.x; Microsoft Internet Explorer 2.x, 3.x, 4.x.

Exemplu:

```
<HTML>
  <HEAD>
    <TITLE>Heading examples</TITLE>
  </HEAD>
    <H1> This little piggy is heading style 1</H1>
    <H2> This little piggy is heading style 2</H2>
    <H3> This little piggy is heading style 3</H3>
    <H4> This little piggy is heading style 4</H4>
    <H5> This little piggy is heading style 5</H5>
    <H6> This little piggy is heading style 6</H6>
    <P> Elementele antet creeaza automat spatiu vertical alb
        ➥care înconjoara uniform antetul.
</HTML>
```

<HR>

Utilizare: Se pot folosi linii orizontale pentru a separa anumite secțiuni dintr-o pagină Web.

Sintaxa: `<HR[atribute]>`

Eticheta de start/stop: Cerută/Interzisă

Atribute: `align = "..."` Este nerecomandat. Controlează alinierea (`left, center, right, justify`).

`noshade = "..."` Afișează linia despărțitoare cu o culoare plină.

`size = "..."` Este nerecomandat. Controlează dimensiunea liniei despărțitoare.

`width = "..."` Este nerecomandat. Controlează lățimea liniei despărțitoare.

`class=, id=, style=, title=, lang=, dir=` Atribute uzuale.

Intrinseci: `onclick=, ondblclick=, onmousedown=, onmouseup=, onmouseover=, onmousemove=, onmouseout=, onkeypress=, onkeydown=, onkeyup=` Evenimente intrinseci uzuale.

Gol: Da, eticheta de încheiere este interzisă.

Observații: Există diverse modalități de folosire a etichetei `<HR>` , dar unii creatori se feresc de această etichetă pentru a folosi mai bine spațiul liber pentru a separa elementele din pagină.

Introdus: HTML 2.0

Browsere: Netscape Navigator 0.9, 1.x, 2.x, 3.x, 4.x; Microsoft Internet Explorer 2.x, 3.x, 4.x.

Exemplu:

```
...
  <H2>Capitolul unu</H2>
...
<HR>
  <H2>Capitolul doi</H2>
...
```

<ISINDEX>* °

Observații: Operează ca și un element la nivel de bloc care creează un formular simplu. A fost prezentat în secțiunea „Elemente structurale (de organizare)", ca fiind un element de titlu. Este nerecomandat. Folosiți elementul INPUT (intrare) într-un element FORM (formular) în locul lui.

<P>...[</P>]

Utilizare: Definește un paragraf.

Sintaxa: `<P [atribute]>text paragraf[</P>]`

Eticheta de start/stop: Cerută/Opțională

Atribute: `align= "..."` Este nerecomandat. Controlează alinierea (`left`, `center`, `right`, `justify`).

`class=, id=, style=, title=, lang=, dir=` Atribute uzuale.

Intrinseci: `onclick=, ondblclick=, onmousedown=, onmouseup=, onmouseover=, onmousemove=, onmouseout=, onkeypress=, onkeydown=, onkeyup=` Evenimente intrinseci uzuale.

Gol: Nu

Observații: Eticheta de încheiere este de obicei omisă.

Introdus: HTML 1.0

Browsere: Netscape Navigator 1.0, 2.x, 3.x, 4.x; Microsoft Internet Explorer 2.0, 3.0, 4.0.

Exemplu:

```
<P align=center>Limbajul HTML 4 este o modernizare
➥importanta a limbajului HTML standard.
```

<PRE>...[</PRE>]

Utilizare: Afișează textul preformatat.

Sintaxa: `<PRE [atribute]>textul formatat</PRE>`

Eticheta de start/stop: Cerută/ Cerută

Atribute: `width = "..."` Definește lățimea textului formatat.

`class=, id=, style=, title=, lang=, dir=` Atribute uzuale.

Intrinseci: `onclick=, ondblclick=, onmousedown=, onmouseup=, onmouseover=, onmousemove=, onmouseout=, onkeypress=, onkeydown=, onkeyup=` Evenimente intrinseci uzuale.

Gol: Nu

Observații: Elementul <PRE> reprezintă o modalitate excelentă de a adăuga într-o pagină Web documente cu text deja formatat, deoarece păstrează formatarea originală a documentului.

Introdus: HTML 2.0

Browsere: Netscape Navigator 1.0, 2.0, 3.x, 4.x; Microsoft Internet Explorer 2.0, 3.0, 4.0.

Exemplu:

```
<PRE width="80%">
Acquisitions Editors   Technical Editors    Editorial Assistant
 Development Editors   Office Manager       Mailroom Attendant
  Production Editors   Typist
       Copy Editors    Clerk                Other Workers
</PRE>
```

Listele: DD, DIR°, DL, DT, LI, MENU°, OL, UL

La fel ca toate elementele de bloc, elementele listă determină o întrerupere a fluxului textului, dar ele sunt speciale prin faptul că pot conține alte elemente de bloc sau pot fi parte integrantă a altor elemente de bloc. Puteți organiza textul într-o structură mai bine organizată, prin crearea listelor, care la rândul lor pot fi imbricate.

<DD>...[</DD>]

Utilizare: Redă descrierea definiției folosită într-un element DL (lista de definire).

Sintaxa: `<DD [atribute]>descrierea cuvantului care este definit[</DD>]`

Eticheta de start/stop: Cerută/Opțională

Atribute: `class=, id=, style=, title=, lang=, dir=` Atribute uzuale.

Intrinseci: `onclick=, ondblclick=, onmousedown=, onmouseup=, onmouseover=, onmousemove=, onmouseout=, onkeypress=, onkeydown=, onkeyup=` Evenimente intrinseci uzuale.

Gol: Nu

Observații: Poate conține elemente la nivel de bloc, cum ar fi de exemplu elementul <P>.

Introdus: HTML 2.0

Browsere: Netscape Navigator 1.0, 2.0, 3.x, 4.x; Microsoft Internet Explorer 2.0, 3.0, 4.0

Exemplu:

```
<DL>
  <DT> Publications Manager
    <DD> The person in charge of coordinating all
         ➥departamental activities.
  <DT> Technical Writer
    <DD> The person who writes a technical manual.
  <DT> Technical Editor
    <DD> The person who is responsible for the technical
         ➥accuracy of the finished manual.
</DL>
```

<DIR>...</DIR> °

Utilizare: Creează o listă de directoare multicoloană (teoretic).

Sintaxa: `<DIR [atribute]>continutul directorului de liste</DIR>`

Eticheta de start/stop: Cerută/Cerută

Atribute: `compact` Este nerecomandat. Poate afișa lista într-o formulă compactă.

`class=, id=, style=, title=, lang=, dir=` Atribute uzuale.

Intrinseci: `onclick=, ondblclick=, onmousedown=, onmouseup=, onmouseover=, onmousemove=, onmouseout=, onkeypress=, onkeydown=, onkeyup=` Evenimente intrinseci uzuale.

Gol: Nu

Observații: Trebuie să conțină cel puțin un element în listă. Acest element este nerecomandat, în favoarea elementului `UL` (lista neordonată). Inițial, acest element a fost folosit pentru listarea fișierelor.

Introdus: HTML 2.0

Browsere: Netscape Navigator 1.0, 2.0, 3.x, 4.x; Microsoft Internet Explorer 2.0, 3.0, 4.0.

Exemplu:

```
<DIR>
  <LI><CODE>README.</CODE>
  <LI><CODE>INSTALL</CODE>
  <LI><CODE>ERRATA</CODE>
</DIR>
```

<DL>...</DL>

Utilizare: Creează o listă de definiri.

Sintaxa: `<DL [atribute]>continutul listei de definiri</DL>`

Eticheta de start/stop: Cerută/Cerută

Atribute: `compact` Este nerecomandat. Poate afișa lista într-o formulă compactă.

`class=, id=, style=, title=, lang=, dir=` Atribute uzuale.

Intrinseci: `onclick=, ondblclick=, onmousedown=, onmouseup=, onmouseover=, onmousemove=, onmouseout=, onkeypress=, onkeydown=, onkeyup=` Evenimente intrinseci uzuale.

Gol: Nu

Observații: Trebuie să conțină cel puțin un element `<DT>` sau `<DD>` în orice ordine.

Introdus: HTML 2.0

Browsere: Netscape Navigator 1.0, 2.0, 3.x, 4.x; Microsoft Internet Explorer 2.0, 3.0, 4.0

Exemplu:

```
<DL>
  <DT> Publications Manager
    <DD> The person in charge of coordinating all
         ➥departamental activities.
  <DT> Technical writer
    <DD> The person who writes a technical manual.
  <DT> Technical Editor
    <DD> The person who is responsible for the technical
         ➥accuracy of the finished manual.
</DL>
```


<DT>...</DT>

Utilizare: Termenul de definire (sau eticheta) folosit într-un element `DL` (lista de definiri).

Sintaxa: `<DT [atribute]>textul termenului de definire[<DT>]`

Eticheta de start/stop: Cerută/Opțională

Atribute: `class=, id=, style=, title=, lang=, dir=` Atribute uzuale.

Intrinseci: `onclick=, ondblclick=, onmousedown=, onmouseup=, onmouseover=, onmousemove=, onmouseout=, onkeypress=, onkeydown=, onkeyup=` Evenimente intrinseci uzuale.

Gol: Nu

Observații: Trebuie să conțină text (ce poate fi modificat de elementele de marcare a textului).

Introdus: HTML 2.0

Browsere: Netscape Navigator 1.0, 2.0, 3.x, 4.x; Microsoft Internet Explorer 2.0, 3.0, 4.0

Exemplu:

```
<DL>
  <DT> Publications Manager
    <DD> The person in charge of coordinating all
         ➥departamental activities.
  <DT> Technical writer
    <DD> The person who writes a technical manual.
  <DT> Technical Editor
    <DD> The person who is responsible for the technical
         ➥accuracy of the finished manual.
</DL>
```

<LI>...</LI>

Utilizare: Definește un articol de listă dintr-o listă.

Sintaxa: `<LI [atribute]>listeaza textul articolului[</LI>]`

Eticheta de start/stop: Cerută/Opțională

Atribute: `type="..."` Schimbă modul de numerotare (`1`, `a`, `A`, `i`, `I`) în listele ordonate sau stilul marcajului de listă (`disc` (disc), `square` (pătrat), `circle` (cerc)) în liste neordonate.

`value="..."` Stabilește numerotarea după un număr întreg dat, începând cu articolul curent din listă.

`class=, id=, style=, title=, lang=, dir=` Atribute uzuale.

Intrinseci: `onclick=, ondblclick=, onmousedown=, onmouseup=, onmouseover=, onmousemove=, onmouseout=, onkeypress=, onkeydown=, onkeyup=` Evenimente intrinseci uzuale.

Gol: Nu

Introdus: HTML 2.0

Browsere: Netscape Navigator 1.0, 2.0, 3.x, 4.x; Microsoft Internet Explorer 2.0, 3.0, 4.0

Exemplu:
```
<UL>
    <LI>Articolul 1
    <LI>Articolul 2
</UL>
```

<MENU>...</MENU>*

Utilizare: Creează o listă de meniuri structurată pe o singură coloană.

Sintaxa: `<MENU [atribute]>textul meniului</MENU>`

Eticheta de start/stop: Cerută/Cerută

Atribute: `compact` Este nerecomandat. Poate afișa lista într-o formulă compactă.

`class=, id=, style=, title=, lang=, dir=` Atribute uzuale.

Intrinseci: `onclick=, ondblclick=, onmousedown=, onmouseup=, onmouseover=, onmousemove=, onmouseout=, onkeypress=, onkeydown=, onkeyup=` Evenimente intrinseci uzuale.

Gol: Nu.

Observații: Trebuie să conțină cel puțin un element de listă. Acest element este nerecomandat în favoarea elementului `UL` (listă neordonată).

Introdus: HTML 3.2

Browsere: Netscape Navigator 1.0, 2.0, 3.x, 4.x; Microsoft Internet Explorer 2.0, 3.0, 4.0

Exemplu:
```
...
  <MENU>
    <LI>Articolul 1
    <LI>Articolul 2
  </MENU>
...
```

<OL>...</ OL>

Utilizare: Creează o listă ordonată.

Sintaxa: `<OL [atribute]>textul listei ordonate</OL>`

Eticheta de start/stop: Cerută/Cerută

Atribute: `type="..."` Stabilește stilul numerotării (`1, a, A, i, I`).

`compact` Este nerecomandat. Poate afișa lista într-o formulă compactă.

`start="..."` Stabilește numărul de pornire la numărul întreg ales.

`class=, id=, style=, title=, lang=, dir=` Atribute uzuale.

Intrinseci: `onclick=, ondblclick=, onmousedown=, onmouseup=, onmouseover=, onmousemove=, onmouseout=, onkeypress=, onkeydown=, onkeyup=` Evenimente intrinseci uzuale.

Gol: Nu

Observații: Trebuie să conțină cel puțin un element de listă.

Introdus: HTML 2.0

Browsere: Netscape Navigator 1.0, 2.0, 3.x, 4.x; Microsoft Internet Explorer 2.0, 3.0, 4.0.

Exemplu:
```
<OL>
  <LI>Introduceti CD-ul în unitatea CD-ROM.
  <LI>Alegeti optiunea Run din meniul Start.
  <LI>Introduceti<code>f:setup</code> pentru a începe
     ➥instalarea.
</OL>
```

<UL>...</UL>

Utilizare: Creează o listă neordonată.

Sintaxa: `<UL [atribute]>textul listei neordonate</UL>`

Eticheta de start/stop: Cerută/Cerută

Atribute: `type="..."` Stabilește stilul marcajului de listă (`disc, square, circle`).

`compact` Este nerecomandat. Poate afișa lista într-o formulă compactă.

`class=, id=, style=, title=, lang=, dir=` Atribute uzuale.

Intrinseci: `onclick=, ondblclick=, onmousedown=, onmouseup=, onmouseover=, onmousemove=, onmouseout=, onkeypress=, onkeydown=, onkeyup=` Evenimente intrinseci uzuale.

Gol: Nu

Observații: Trebuie să conțină cel puțin un element de listă.

Introdus: HTML 2.0

Browsere: Netscape Navigator 1.0, 2.0, 3.x, 4.x; Microsoft Internet Explorer 2.0, 3.0, 4.0.

Exemplu:
```
<UL>
  <LI>Articolul 1
  <LI>Articolul 2
  <LI>Articolul 3
</UL>
```

Formulare: BUTTON, FIELDSET, FORM, INPUT, ISINDEX*, LABEL, LEGEND, OPTION, SELECT, TEXTAREA

La fel ca și alte elemente de bloc, formularele determină o întrerupere a fluxului textului și a formatării. Formularele reprezintă un caz special deoarece ele pot conține sau pot fi incluse în alte elemente de bloc și creează o interfață pentru ca utilizatorul să selecteze opțiunile și să trimită datele spre serverul Web.

<BUTTON>...</BUTTON>

Utilizare: Creează un buton.

Sintaxa: `<BUTTON [atribute]>butonul</BUTTON>`

Eticheta de start/stop: Cerută/Cerută

Atribute:
- `name="..."` Reprezintă numele butonului.
- `value="..."` Reprezintă valoarea butonului.
- `type="..."` Reprezintă tipul butonului (`button` (buton), `submit` (trimitere), `reset` (reinițializare)).
- `disabled="..."` Stabilește starea butonului în poziția dezactivat.
- `tabindex="..."` Stabilește ordinea de tabulare între elemente cu un `tabindex` precizat.
- `class=, id=, style=, title=, lang=, dir=` Atribute uzuale.

Intrinseci:
- `onfocus="..."` Evenimentul care apare atunci când elementului i se atribuie focusul.
- `onblur="..."` Evenimentul care apare atunci când elementul pierde focusul.
- `onclick=, ondblclick=, onmousedown=, onmouseup=, onmouseover=, onmousemove=, onmouseout=, onkeypress=, onkeydown=, onkeyup=` Evenimente intrinseci uzuale.

Gol: Nu

Introdus: HTML 3.2

Browsere: Netscape Navigator 3.x, 4.x; Microsoft Internet Explorer 3.0, 4.0.

Exemplu:

```
<FORM action="..."method="...">
  <INPUT type="radio" name= sex value="M"> Male
  <INPUT type="radio" name= sex value="F"> Female
  <BUTTON type="submit" name="submit" value="submit">
    <IMG src="/icons/submit.gif" alt="submit"></BUTTON>
</FORM>
```

<FIELDSET>...</FIELDSET>

Utilizare: Grupează controalele înrudite.

Sintaxa: `<FIELDSET [atribute]>controalele inrudite</FIELDSET>`

Eticheta de start/stop: Cerută/Cerută

Gol: Nu

Introdus: HTML 3.2

Browsere: Netscape Navigator 3.x, 4.x; Microsoft Internet Explorer 3.0, 4.0.

Exemplu:

```
<HTML>
  <HEAD>
    <TITLE>Fieldset Legend</TITLE>
  </HEAD>
  <BODY>
    <FORM action="http://www.example.com/cgi-bin/testing.pl"
    ➥method="post">
      <FIELDSET>
        <LEGEND align="top">Customer Name</LEGEND>
        Last name: <INPUT name="lastname" type="text"
        ➥tabindex="1">
        First name: <INPUT name="firstname" type="text"
        ➥tabindex="2">
      </FIELDSET>
      <FIELDSET>
        <LEGEND align="top">Interests</LEGEND>
        <INPUT name="surfing"
               type="checkbox"
               value="surfing" tabindex="3">Surfing</INPUT>
        <INPUT name="hiking"
               type="checkbox"
               value="hiking"tabindex="4">Hiking</INPUT>
      </FIELDSET>
    </FORM>
  </BODY>
</HTML>
```


<FORM>...</ FORM>

Utilizare: Creează un formular care conține control pentru intrări de la utilizator.

Sintaxa: `<FORM [atribute]>continutul formularului</FORM>`

Eticheta de start/stop: Cerută/Cerută.

Atribute:
- `action="..."` Reprezintă adresa URL pentru acțiunea serverului.
- `method="..."` Reprezintă metoda HTTP (`GET`, `POST`). Eticheta `GET` este nerecomandată.
- `enctype="..."` Precizează tipul MIME (tipul mediului Internet).
- `target="..."` Stabilește dacă va fi afișată resursa. (nume definit de utilizator, `_blank`,`_parent`,`_self`,`_top`).
- `accept-charset="..."` Reprezintă lista codurilor de caracter. Valoarea prestabilită este `ISO-8859-1`.
- `class=, id=, style=, title=, lang=, dir=` Atribute uzuale.

Intrinseci:
- `onsubmit="..."` Evenimentul intrinsec ce apare atunci când este trimis formularul.
- `onreset="..."` Evenimentul intrinsec ce apare atunci când formularul este reinițializat.
- `onclick=, ondblclick=, onmousedown=, onmouseup=, onmouseover=, onmousemove=, onmouseout=, onkeypress=, onkeydown=, onkeyup=` Evenimente intrinseci uzuale.

Gol: Nu

Introdus: HTML 2.0

Browsere: Netscape Navigator 1.0, 2.x, 3.x, 4.x; Microsoft Internet Explorer 2.0, 3.0, 4.0.

Exemplu:
```
<FORM action="http://www.example.com/cgi-bin/register.pl">
 contents of form
</FORM>
```

<INPUT>

Utilizare: Definește controalele folosite în formular.

Sintaxa: `<INPUT [atribute]>`

Eticheta de start/stop: Cerută/Interzisă

Atribute:

`type="..."` Definește tipul controlului de intrare (`text` (text), `password` (parolă), `checkbox` (casetă de validare), `radio` (buton radio), `submit` (trimitere), `reset` (reinițializare), `file` (fisier), `hidden` (ascuns), `image` (imagine), `button` (buton).

`accept="..."` Reprezintă tipurile de fișiere permise pentru descărcare.

`name="..."` Reprezintă numele controlului (cerut pentru toate tipurile de controale, cu excepția butoanelor de tip `submit` și `reset`).

`value="..."` Definește valoarea inițială a controlului (cerută pentru butoanele radio și casetele de validare).

`checked="..."` Stabilește butoanele radio la o stare marcată.

`disabled="..."` Dezactivează controlul.

`readonly="..."` Pentru tipurile de parole text.

`size="..."` Definește lățimea controlului în pixeli, cu excepția controalelor pentru text și parolă, care sunt date în numere de caractere.

`maxlength="..."` Reprezintă numărul maxim de caractere ce pot fi introduse.

`src="..."` Reprezintă adresa URL spre un tip de control al imaginii.

`alt="..."` Reprezintă o descriere alternativă în text.

`usemap="..."` Reprezintă adresa URL spre o hartă de imagini pe partea clientului.

`align="..."` Este nerecomandat. Controlează alinierea (`left`, `center`, `right`, `justify`).

`tabindex="..."` Stabilește ordinea de tabulare între elemente, cu un tabindex definit.

`class=, id=, style=, title=, lang=, dir=` Atribute uzuale.

Intrinseci:

`onfocus="..."` Evenimentul care apare atunci când elementului i se atribuie focusul.

`onblur="..."` Evenimentul care apare atunci când elementul își pierde focusul.

`onselect="..."` Eveniment intrinsec, care apare atunci când este selectat controlul.

`onchange="..."` Eveniment intrinsec, care apare atunci când controlul este modificat.

`onclick=, ondblclick=, onmousedown=, onmouseup=, onmouseover=, onmousemove=, onmouseout=, onkeypress=, onkeydown=, onkeyup=` Evenimente intrinseci uzuale.

BIBL. CENTR. UNIV. „M. EMINESCU" IAȘI INFORMATICĂ

Gol: Da, eticheta de încheiere este interzisă.

Tipurile de intrări:

`text` Creează o casetă de text cu o singură linie.

`password` Este similară cu introducerea de text, dar textul introdus este redat în câmp într-un mod care ascunde caracterele alfanumerice introduse.

`checkbox` Este un comutator on/off. Când este marcat este activat (on), când nu este marcat, este dezactivat (off).

`radio` Este un comutator de tip închis/deschis.

`submit` Creează un buton de tip submit pentru textul ce trebuie transmis.

`image` Creează un buton grafic de tip submit.

`reset` Creează un buton de reinițializare.

`button` Creează un buton fără o comportare prestabilită.

`hidden` Creează un element care nu este redat de agentul pentru utilizator.

`file` Cere utilizatorului să introducă un nume de fișier.

Introdus: HTML 2.0

Browsere: Netscape Navigator 1.0, 2.x, 3.x, 4.x; Microsoft Internet Explorer 2.0, 3.0, 4.0.

Exemplu:
```
<FORM ...>
  <INPUT type="radio" name="sex" value="M"> Male
  <INPUT type="radio" name="sex" value="F"> Female
</FORM>
```

<ISINDEX>* [0]

Observații: Operează ca un element la nivel de bloc, care creează un formular simplu. A fost discutat în secțiunea „Elemente structurale (de organizare)" ca fiind un element de titlu. Este nerecomandat. Folosiți în locul lui elementul `INPUT` într-un formular (`FORM`).

<LABEL>...</LABEL>

Utilizare: Etichetează un control.

Sintaxa: `<LABEL[atribute]>controlul etichetat</LABEL>`

Eticheta de start/stop: Cerută/Cerută

Atribute:
`for="..."` Asociază o etichetă unui control identificat.
`disabled="..."` Dezactivează controlul.
`accesskey="..."` Asociază o tastă rapidă acestui element.
`class=, id=, style=, title=, lang=, dir=` Atribute uzuale.

Intrinseci:
`onfocus="..."` Definește evenimentul care apare atunci când elementului i se atribuie focusul.
`onblur="..."` Definește evenimentul care apare atunci când elementul își pierde focusul.
`onclick=, ondblclick=, onmousedown=, onmouseup=, onmouseover=, onmousemove=, onmouseout=, onkeypress=, onkeydown=, onkeyup=` Evenimente intrinseci uzuale.

Gol: Nu

Introdus: HTML 4.0

Browsere: Netscape Navigator 4.x; Microsoft Internet Explorer 4.0.

Exemplu:

```
<HTML>
  <FORM action="..." method="post">
    <TABLE>
     <TR>
       <TD><LABEL for="first">First name</LABEL>
          <TD><INPUT type="text" name="firstname"
          ➥id="first">
     <TR>
       <TD><LABEL for="last">Last name</LABEL>
          <TD><INPUT type="text" name="lastname" id="last">
...
    </TABLE>
  </FORM>
</HTML>
```

<LEGEND>...</LEGEND>

Utilizare: Asociază un titlu unui grup de controale (`FIELDSET`).

Sintaxa: `<LEGEND [atribute]>titlul grupului de controale</LEGEND>`

Eticheta de start/stop: Cerută/Cerută

Atribute:
`class, id, style, title, lang, dir, onclick, ondblclick, onmousedown, onmouseup, onmouseover, onmousemove, onmouseout, onkeypress, onkeydown, onkeyup`
`align="..."` Este nerecomandat. Controlează alinierea (`left, center, right, justify`).
`accesskey="..."` Asociază o tastă rapidă acestui element.

Gol: Nu

Introdus: HTML 3.2

Browsere: Netscape Navigator 3.x, 4.x; Microsoft Internet Explorer 3.0, 4.0.

Exemplu:

```
<HTML>
  <HEAD>
    <TITLE>Fieldset Legend</TITLE>
  </HEAD>
  <BODY>
    <FORM action="http://www.example.com/cgi-bin/testing.pl"
    ➥method="post">
       <FIELDSET>
         <LEGEND align="top">Customer Name</LEGEND>
         Last name: <INPUT name="lastname" type="text"
         ➥tabindex="1">
         First name: <INPUT name="firstname" type="text"
         ➥tabindex="2">
       </FIELDSET>
       <FIELDSET>
         <LEGEND align="top">Interests</LEGEND>
         <INPUT name="surfing"
                type="checkbox"
                value="surfing"tabindex="3">Surfing</INPUT>
         <INPUT name="hiking"
                type="checkbox"
                value="hiking"tabindex="4">Hiking</INPUT>
       </FIELDSET>
     </FORM>
  </BODY>
</HTML>
```

<OPTION>...</OPTION>

Utilizare: Precizează variantele într-un element `SELECT`.

Sintaxa: `<OPTION [atribute]>variantele dintr-un element SELECT</OPTION>`

Eticheta de start/stop: Cerută/Opțională

Atribute:
`selected="..."` Specifică dacă este selectată opțiunea.
`disabled="..."` Dezactivează controlul.
`value="..."` Valoarea trimisă dacă este selectat controlul.
`class=, id=, style=, title=, lang=, dir=` Atribute uzuale.

Intrinseci:
`onclick=, ondblclick=, onmousedown=, onmouseup=, onmouseover=, onmousemove=, onmouseout=, onkeypress=, onkeydown=, onkeyup=` Evenimente intrinseci uzuale.

Gol: Nu

Introdus: HTML 2.0

Browsere: Netscape Navigator 1.0, 2.x, 3.x, 4.x; Microsoft Internet Explorer 2.0, 3.0, 4.0.

Exemplu:

```
<HTML>
  ...
  <FORM action="..."method="post">
     ...
     <SELECT name="order"size="1">
        <OPTION value="one">   One potato
        <OPTION value="two">   Two potato
        <OPTION value="three"> Three potato
     </SELECT>
     ...
  </FORM>
  ...
</HTML>
```

<SELECT>...</SELECT>

Utilizare: Creează variantele de selecție pentru utilizator.

Sintaxa: `<SELECT [atribute]>variantele</SELECT>`

Eticheta de start/stop: Cerută/Cerută

Atribute:

`name ="..."` Reprezintă numele elementului.

`size="..."` Reprezintă lățimea, dată în număr de rânduri.

`multiple` Permite selecții multiple.

`disabled="..."` Dezactivează controlul.

`tabindex="..."` Stabilește ordinea de tabulare între elemente, cu un `tabindex` definit.

`class=, id=, style=, title=, lang=, dir=` Atribute uzuale.

Intrinseci:

`onfocus="..."` Reprezintă evenimentul care apare atunci când elementului i se atribuie focusul.

`onblur="..."` Reprezintă evenimentul care apare atunci când elementul își pierde focusul.

`onselect="..."` Este un eveniment intrinsec, care apare atunci când este selectat controlul.

`onchange="..."` Este un eveniment intrinsec, care apare atunci când controlul este modificat.

`onclick=, ondblclick=, onmousedown=, onmouseup=, onmouseover=, onmousemove=, onmouseout=, onkeypress=, onkeydown=, onkeyup=` Evenimente intrinseci uzuale.

Gol: Nu

Introdus: HTML 3.2

Browsere: Netscape Navigator 1.0, 2.x, 3.x, 4.x; Microsoft Internet Explorer 3.0, 4.0.

Exemplu:

```
<HTML>
  ...
  <FORM action="..."method="post">
     ...
  <FORM action="..."method="post">
     ...
     <SELECT name="choice" size="2">
        <OPTION value="one">One fish
        <OPTION value="two">Two fish
        <OPTION value="red">Red fish
        <OPTION value="blue">Blue fish
     </SELECT>
     ...
  </FORM>
  ...
</HTML>
```


<TEXTAREA>...</ TEXTAREA>

Utilizare: Creează o suprafață cu linii multiple pentru intrarea de la utilizator.

Sintaxa: `<TEXTAREA [atribute]>suprafata pentru intrarea de la utilizator</TEXTAREA>`

Eticheta de start/stop: Cerută/Cerută

Atribute:

`name="..."` Definește numele controlului.

`rows="..."` Definește lățimea, dată în număr de rânduri.

`cols="..."` Definește înălțimea, dată în număr de coloane.

`disabled="..."` Dezactivează controlul.

`readonly="..."` Stabilește textul afișat în modul protejat la scriere.

`tabindex="..."` Stabilește ordinea de tabulare între elemente, cu un `tabindex` definit.

`class=, id=, style=, title=, lang=, dir=` Atribute uzuale.

Intrinseci:

`onfocus="..."` Reprezintă evenimentul care apare atunci când elementului i se atribuie focusul.

`onblur="..."` Reprezintă evenimentul care apare atunci când elementul își pierde focusul.

`onselect="..."` Este un eveniment intrinsec, care apare atunci când este selectat controlul.

`onchange="..."` Este un eveniment intrinsec, care apare atunci când controlul este modificat.

`onclick=, ondblclick=, onmousedown=, onmouseup=, onmouseover=, onmousemove=, onmouseout=, onkeypress=, onkeydown=, onkeyup=` Evenimente intrinseci uzuale.

Gol: Nu

Observații: Textul care trebuie afișat este plasat între etichetele de început și de sfârșit.

Introdus: HTML 3.2

Browsere: Netscape Navigator 1.0, 2.x, 3.x, 4.x; Microsoft Internet Explorer 3.0, 4.0.

Exemplu:

```
<HTML>
  <HEAD>
    <TITLE>Form Demonstration</TITLE>
  </HEAD>
  <BODY>
    <FORM="http://www.example.com/sender" method="post">
      <TEXTAREA rows="30" cols="60">
        Continut zona de text preincarcat
      </TEXTAREA>
  </BODY>
</HTML>
```

Tabele: CAPTION, COL, COLGROUP, TABLE, TBODY, TD, TFOOT, THEAD, TR

Tabelele sunt destinate să afișeze datele într-un format tabular. Înainte de introducerea limbajului HTML 4.0, tabelele au fost foarte folosite în scopuri de aranjare în pagină, dar o dată cu apariția foilor de stil, Consorțiul W3C descurajează folosirea lor în acest scop.

<CAPTION>...</CAPTION>

Utilizare: Afișează capul de tabel.

Sintaxa: `<CAPTION [atribute]>textul capului de tabel</CAPTION>`

Eticheta de start/stop: Cerută/Cerută

Atribute: `align="..."` Este nerecomandat. Controlează alinierea (`left`, `center`, `right`, **`top`**, `bottom`).

`class=, id=, style=, title=, lang=, dir=` Atribute uzuale.

Intrinseci: `onclick=, ondblclick=, onmousedown=, onmouseup=, onmouseover=, onmousemove=, onmouseout=, onkeypress=, onkeydown=, onkeyup=` Evenimente intrinseci uzuale.

Gol: Nu

Observații: Este opțional.

Introdus: HTML 3.2

Browsere: Netscape Navigator 1.1, 2.0, 3.x, 4.x; Microsoft Internet Explorer 2.0, 3.0, 4.0.

Exemplu:

```
  <TABLE cellspacing="2" cellpadding="5">
<!-- Use IE valign= in addition to align= to achieve
    ➥the same --
  -- effect on the page in spite of IE non-standard
    ➥syntax -->
  <CAPTION align="bottom" valign="bottom">Holmes, 1997
  ➥</CAPTION>
```


<COL>

Utilizare: Grupează coloanele pe grupuri de coloane pentru a partaja valorile atributelor.

Sintaxa: `<COL [atribute]>`

Eticheta de start/stop: Cerută/Interzisă

Atribute: `span="..."` Definește numărul de coloane pe care le conține grupul.

`width="..."` Definește lățimea coloanei dată în procente, pixeli sau valoare minimă.

`align="..."` Aliniază orizontal conținutul celulelor (**`left`**, `center`, `right`, `justify`, `char`).

`char="..."` Stabilește un anumit caracter după care se aliniază coloana. Valoarea prestabilită este caracterul punct zecimal din limba prestabilită.

`charoff="..."` Reprezintă distanța față de primul caracter de aliniere dintr-o linie.

`valign="..."` Aliniază pe verticală conținutul unei celule (`top`, `middle`, `bottom`, `baseline`).

`class=, id=, style=, title=, lang=, dir=` Atribute uzuale.

Intrinseci: `onclick=, ondblclick=, onmousedown=, onmouseup=, onmouseover=, onmousemove=, onmouseout=, onkeypress=, onkeydown=, onkeyup=` Evenimente intrinseci uzuale.

Gol: Da, eticheta de încheiere este interzisă.

Introdus: HTML 4.0

Browsere: Netscape Navigator 3.x, 4.x; Microsoft Internet Explorer 2.0, 3.0, 4.0.

Exemplu:

```
<TABLE cellspacing="2" cellpadding="5">
  <COLGROUP>
     <COL width="40">
     <COL width="0*">
     <COL width="2*">
  <COLGROUP align="center">
     <COL width="1*">
     <COL width="3*" align="char" char=":">
  <THEAD>
    <TR>
    ...
  </TABLE>
```

<COLGROUP>...[</COLGROUP>]

Utilizare: Definește un grup de coloane.

Sintaxa: `<COLGROUP [atribute]>grupul de coloane[</COLGROUP>]`

Eticheta de start/stop: Cerută/Opțională

Atribute: `span="..."` Reprezintă numărul de coloane dintr-un grup.

`width="..."` Reprezintă lățimea coloanelor.

`align="..."` Aliniază pe orizontală conținutul celulelor (**`left`**, `center`, `right`, `justify`, `char`).

`char="..."` Stabilește un anumit caracter după care se aliniază coloana. Valoarea prestabilită este caracterul punct zecimal din limba prestabilită.

`charoff="..."` Reprezintă distanța față de primul caracter de aliniere dintr-o linie.

`valign="..."` Aliniază pe verticală conținutul unei celule (`top`, **`middle`**, **`bottom`**, `baseline`).

`cellpadding="..."` Definește spațierea în celule.

`class=, id=, style=, title=, lang=, dir=` Atribute uzuale.

Intrinseci: `onclick=, ondblclick=, onmousedown=, on mouseup=, onmouseover=, onmousemove=, onmouseout=, onkeypress=, onkeydown=, onkeyup=` Evenimente intrinseci uzuale.

Gol: Nu

Introdus: HTML 3.2

Browsere: Netscape Navigator 3.x, 4.x; Microsoft Internet Explorer 2.0, 3.0, 4.0.

Exemplu:

```
<TABLE>
<COLGROUP span="2" width="200">
  <TR>
     <TH>Microsoft
        <TH>Corel
  <TR>
     <TD>MS Word 97
        <TD>WordPerfect 8
</TABLE>
```

<TABLE>....</TABLE>

Utilizare: Definește un tabel.

Sintaxa: `[<TABLE [atribute]>]continutul tabelului[</TABLE>]`

Eticheta de start/stop: Cerută/Cerută

Atribute: `align="..."` Aliniază orizontal conținutul celulelor (**`left`**, `center`, `right`, `justify`, `char`).

`bgcolor="..."` Este nerecomandat. Stabilește culoarea fundalului pentru tabel.

`border="..."` Reprezintă lățimea bordurii care înconjoară tabelul, dată în pixeli.

`cellpading="..."` Definește spațiul care înconjoară fiecare celulă.

`cellspacing="..."` Definește spațiul din interiorul fiecărei celule.

`cols="..."` Definește numărul de coloane din tabel.

`frame="..."` Definește care parte a cadrului tabelului va fi vizibil (**`void`** (gol), `above` (deasupra), `below` (dedesubt), `hsides`, `lhs`, `vsides`, `box` (casetă), `border` (bordură)).

`rules="..."` Definește care dintre liniile din interiorul tabelului vor fi vizibile (**none (nici una)**, `groups` (grupuri), `rows` (rânduri), `cols` (coloane), `all` (toate)).

`width="..."` Definește lățimea dorită pentru tabel.

`class=, id=, style=, title=, lang=, dir=` Atribute uzuale.

Intrinseci: `onclick=, ondblclick=, onmousedown=, onmouseup=, onmouseover=, onmousemove=, onmouseout=, onkeypress=, onkeydown=, onkeyup=` Evenimente intrinseci uzuale.

Gol: Nu

Introdus: HTML 3.2

Browsere: Netscape Navigator 3.x, 4.x; Microsoft Internet Explorer 2.0, 3.0, 4.0.

Exemplu:

```
<TABLE>
  <TR>
     <TH>Microsoft
        <TH>Corel
  <TR>
     <TD>MS Word 97
        <TD>WordPerfect 8
</TABLE>
```

[<TBODY>]...[</TBODY>]

Utilizare: Definește corpul tabelului.

Sintaxa: `[<TBODY [atribute]>]continutul corpului de tabel[</TBODY>]`

Eticheta de start/stop: Opțională/Opțională

Atribute: `align="..."` Aliniază orizontal conținutul celulelor (**`left`**, `center`, `right`, `justify`, `char`).

`char="..."` Stabilește un anumit caracter după care se aliniază coloana. Valoarea prestabilită este caracterul punct zecimal din limba prestabilită.

`charoff="..."` Definește distanța față de primul caracter de aliniere dintr-o linie.

`valign="..."` Aliniază pe verticală conținutul celulelor (`top`, `middle`, `bottom`, `baseline`).

`class=, id=, style=, title=, lang=, dir=` Atribute uzuale.

Intrinseci: `onclick=, ondblclick=, onmousedown=, onmouseup=, onmouseover=, onmousemove=, onmouseout=, onkeypress=, onkeydown=, onkeyup=` Evenimente intrinseci uzuale.

Gol: Nu

Introdus: HTML 3.2

Browsere: Netscape Navigator 3.x, 4.x; Microsoft Internet Explorer 2.0, 3.0, 4.0.

Exemplu:

```
<HTML>
<HEAD>
  <TITLE>Flange Manufacturing Statistics</TITLE>
</HEAD>
<BODY>
...
  <TABLE cellspacing="2"cellpadding="5">
    <CAPTION align= "bottom" valign="bottom">Flange usage
             data from Holmes, 1997</CAPTION>
    <THEAD>
      <TR>
        <TH> colspan="4"align="center">
            <B>Flange Market Share by Type</B>
        <TR>
            <TH>Closet
                <TH>Safety
                    <TH>Stanchion
                        <TH>Wheel
    <TBODY>
      <TR>
          <TD>21%
              <TD>48%
                  <TD>12%
                      <TD>73%
    <TFOOT>
      <TR>
          <TD colspan="4" align="center">
              <B>Note: These figures are entirely
              ➥fictional</B>
  </TABLE>
...
</BODY>
</HTML>
```

<TD>...[</ TD>]

Utilizare: Definește conținutul unei celule.

Sintaxa: `[<TD [atribute]>continutul celulei[</TD>]`

Eticheta de start/stop: Cerută /Opțională

Atribute:

`axis="..."` Este numele prescurtat. Valoarea prestabilită este valoarea conținutului celulei.

`axes="..."` Reprezintă una sau mai multe denumiri `axis` care indică titlurile rândurilor și/sau coloanelor referitoare la celulă și sunt separate prin virgule.

`nowrap="..."` Este nerecomandat. Dezactivează funcția de aranjare automată a textului într-o celulă.

`bgcolor="..."` Este nerecomandat. Stabilește culoarea fundalului.

`rowspan="..."` Reprezintă numărul de rânduri pe care se întinde o celulă.

`colspan="..."` Reprezintă numărul de coloane pe care se întinde o celulă.

`align="..."` Aliniază pe orizontală conținutul celulelor (**`left`**, `center`, `right`, `justify`, `char`).

`char="..."` Stabilește un caracter după care se aliniază coloana. Valoarea prestabilită este caracterul punct zecimal din limba prestabilită.

`charoff="..."` Definește distanța față de primul caracter de aliniere dintr-o linie.

`valign="..."` Aliniază pe verticală conținutul celulelor (`top`, **`middle`**, `bottom`, `baseline`).

`class=, id=, style=, title=, lang=, dir=` Atribute uzuale.

Intrinseci: `onclick=, ondblclick=, onmousedown=, onmouseup=, onmouseover=, onmousemove=, onmouseout=, onkeypress=, onkeydown=, onkeyup=` Evenimente intrinseci uzuale.

Gol: Nu

Introdus: HTML 2.0

Browsere: Netscape Navigator 1.1, 2.0, 3.x, 4.x; Microsoft Internet Explorer 2.0, 3.0, 4.0.

Exemplu:

```
<TR>
     <TD>MS Word 97
        <TD>FrameMaker
```

<TFOOT>...[</TFOOT>]

Utilizare: Definește subsolul unui tabel.

Sintaxa: `<TFOOT [atribute]>continutul subsolului</TFOOT>`

Eticheta de start/stop: Cerută /Opțională

Atribute:

`align="..."` Aliniază orizontal conținutul celulelor (**`left`**, `center`, `right`, `justify`, `char`).

`char="..."` Stabilește un caracter după care se aliniază coloana. Valoarea prestabilită este caracterul punct zecimal din limba prestabilită.

`charoff="..."` Definește distanța față de primul caracter de aliniere dintr-o linie.

`valign="..."` Aliniază pe verticală conținutul celulelor (`top`, `middle`, `bottom`, `baseline`).

class=, id=, style=, title=, lang=, dir= Atribute uzuale.

Intrinseci: onclick=, ondblclick=, onmousedown=, onmouseup=, onmouseover=, onmousemove=, onmouseout=, onkeypress=, onkeydown=, onkeyup= Evenimente intrinseci uzuale.

Gol: Nu

Introdus: HTML 2.0

Browsere: Netscape Navigator 2.x, 3.x, 4.x; Microsoft Internet Explorer 2.0, 3.0, 4.0.

Exemplu:

```
<TFOOT>
  <TR>
    <TD colspan="4"
        align="center">Note: Data from Holmes, 1997
```

<TH>...[</TH>]

Utilizare: Definește celulele de date ale antetului tabelului.

Sintaxa: `<TH [atribute]>continutul antetului tabelului</TH>`

Eticheta de start/stop: Cerută / Opțională

Atribute: axis="..." Este numele prescurtat. Valoarea prestabilită este dată de conținutul celulei.

axes="..." Reprezintă una sau mai multe denumiri axis care indică titlurile rândurilor și/sau coloanelor referitoare la celulă și separate prin virgule.

nowrap="..." Este nerecomandat. Dezactivează funcția de aranjare automată a textului într-o celulă.

bgcolor="..." Este nerecomandat. Stabilește culoarea fundalului.

rowspan="..." Reprezintă numărul de rânduri pe care se întinde o celulă.

colspan="..." Reprezintă numărul de coloane pe care se întinde o celulă.

align="..." Aliniază pe orizontală conținutul celulelor (**left**, center, right, justify, char).

char="..." Stabilește un caracter după care se aliniază coloana. Valoarea prestabilită este caracterul punct zecimal din limba prestabilită.

charoff="..." Definește distanța față de primul caracter de aliniere dintr-o linie.

valign="..." Aliniază pe verticală conținutul celulelor (top, **middle**, bottom, baseline).

class=, id=, style=, title=, lang=, dir= Atribute uzuale.

Intrinseci: onclick=, ondblclick=, onmousedown=, onmouseup=, onmouseover=, onmousemove=, onmouseout=, onkeypress=, onkeydown=, onkeyup= Evenimente intrinseci uzuale.

Gol: Nu

Observații: Browserele ar trebui să centreze textul din conținut. Multe browsere de asemenea scriu textul din titlu cu aldine.

Introdus: HTML 2.0

Browsere: Netscape Navigator 1.1, 2.0, 3.x, 4.x; Microsoft Internet Explorer 2.0, 3.0, 4.0.

Exemplu:

```
...
<TABLE>
  <TR>
    <TH>MS Word 97
      <TH>FrameMaker
  <TR>
    <TD>97
      <TD>102
</TABLE>
...
```

<THEAD>...[</THEAD>]

Utilizare: Definește antetul tabelului.

Sintaxa: `<THEAD [atribute]>continutul antetului de tabel</THEAD>`

Eticheta de start/stop: Cerută /Opțională

Atribute: align="..." Aliniază pe orizontală conținutul celulelor (**left**, center, right, justify, char).

char="..." Stabilește un caracter după care se aliniază coloana. Valoarea prestabilită este caracterul punct zecimal din limba prestabilită.

charoff="..." Definește distanța față de primul caracter de aliniere dintr-o linie.

valign="..." Aliniază pe verticală conținutul celulelor (top, middle, bottom, baseline).

class=, id=, style=, title=, lang=, dir= Atribute uzuale.

Intrinseci: onclick=, ondblclick=, onmousedown=, onmouseup=, onmouseover=, onmousemove=, onmouseout=, onkeypress=, onkeydown=, onkeyup= Evenimente intrinseci uzuale.

Gol: Nu

Introdus: HTML 2.0

Browsere: Netscape Navigator 3.x, 4.x; Microsoft Internet Explorer 2.0, 3.0, 4.0.

Exemplu:

```
<HTML>
  <HEAD>
    <TITLE>Tables Demonstration</TITLE>
  </HEAD>
  <BODY>
```

```
<TABLE>
 <CAPTION>This is a caption</CAPTION>
</THEAD>
 <TR>
    <TH>Column One
       <TH>Column Two
<TBODY>
 <TR>
    <TD>Sample A
    <TD>Sample B
<TABLE>
</BODY>
</HTML>
```

<TR>...[</TR>]

Utilizare: Definește un rând de celule din tabel.

Sintaxa: `<TR [atribute]>continutul rândului din tabel</TR>`

Eticheta de start/stop: Cerută /Opțională

Atribute:

`align="..."` Aliniază pe orizontală conținutul celulelor (`left`, `center` `right`, `justify`, `char`).

`char="..."` Stabilește un caracter după care se aliniază coloana. Valoarea prestabilită este caracterul punct zecimal din limba prestabilită.

`charoff="..."` Definește distanța față de primul caracter de aliniere dintr-o linie.

`valign="..."` aliniază pe verticală conținutul celulelor (`top`, `middle`, `bottom`, `baseline`).

`bgcolor="..."` Este nerecomandat. Stabilește culoarea fundalului.

`class=`, `id=`, `style=`, `title=`, `lang=`, `dir=` Atribute uzuale.

Intrinseci: `onclick=`, `ondblclick=`, `onmousedown=`, `onmouseup=`, `onmouseover=`, `onmousemove=`, `onmouseout=`, `onkeypress=`, `onkeydown=`, `onkeyup=` Evenimente intrinseci uzuale.

Gol: Nu

Introdus: HTML 2.0

Browsere: Netscape Navigator 1.1, 2.x, 3.x, 4.x; Microsoft Internet Explorer 2.0, 3.0, 4.0.

Exemplu:

```
<HTML>
  <HEAD>
    <TITLE>File Formats</TITLE>
  </HEAD>
  <BODY>
    <TABLE>
     <TR>
```



```
        <TH>Graphics Formats
           <TH>Doc Formats
     <TR>
        <TD>.gif
           <TD>.doc
     <TR>
        <TD>.jpg
           <TD>.fm
    </TABLE>
  </BODY>
</HTML>
```

Elemente la nivel de text

Elementele la nivel de text se desfășoară la fel ca și textul și nu modifică direct machetarea sau formatarea documentului. Unele dintre aceste etichete afectează imaginea fizică a textului, altele sunt destinate să marcheze textul cu importanță lexicală, cu un anumit înțeles sau rol. De exemplu, atât `<I>` cât și `<EM>` sunt de obicei redate italic, dar `<EM>` se referă la evidențiere, care poate fi disponibilă în limbi ce nu au un echivalent pentru caracterele italice.

Marcatoare de text: B, BIG, I, S⁰, SMALL, STRIKE⁰, TT, U⁰

Anumite caracteristici ale textului, cum ar fi dimensiunea, grosimea și stilul, pot fi modificate direct folosind aceste elemente, dar specificațiile limbajului HTML 4.0 vă încurajează să folosiți foile de stil în locul acestora. Totuși, aceste caracteristici ale textului vor mai rezista încă mult timp, pentru a susține nevoile utilizatorilor de Web care nu aleg browserele „de top" cum ar fi Netscape Communicator 4.0 sau Microsoft Internet Explorer 4.0. Persoanele cu diverse handicapuri, de exemplu, au acces în principal prin folosirea browserelor în mod text, iar multe limbi ne-europene nu prea sunt bine deservite de piața de browsere în general. Toate aceste persoane au nevoie de un alt fel de browser și este bine să fim receptivi și la nevoile lor atunci când realizăm un sit Web.

<B>...</B>

Utilizare: Este folosit pentru a scrie un text cu aldine.

Sintaxa: `<B [atribute]>textul scris cu aldine</B>`

Eticheta de start/stop: Cerută /cerută

Atribute: `class=`, `id=`, `style=`, `title=`, `lang=`, `dir=` Atribute uzuale.

Intrinseci: `onclick=`, `ondblclick=`, `onmousedown=`, `onmouseup=`, `onmouseover=`, `onmousemove=`, `onmouseout=`, `onkeypress=`, `onkeydown=`, `onkeyup=` Evenimente intrinseci uzuale.

Gol: Nu

Introdus: HTML 1.0

Browsere: Netscape Navigator 1.0, 2.0, 3.x, 4.x; Microsoft Internet Explorer 2.0, 3.0, 4.0.

Exemple:

```
<P>The <B>numele serverului virtual</B> ar trebui sa fie
      identic cu numele severului dumneavoastra de Web.
```

<BIG>....</BIG>

Utilizare: Este folosit pentru a mări dimensiunea textului.

Sintaxa: `<BIG [atribute]>textul marit</BIG>`

Eticheta de start/stop: Cerută /Cerută

Atribute: `class=, id=, style=, title=, lang=, dir=` Atribute uzuale

Intrinseci: `onclick=, ondblclick=, onmousedown=, onmouseup=, onmouseover=, onmousemove=, onmouseout=, onkeypress=, onkeydown=, onkeyup=` Evenimente intrinseci uzuale.

Gol: Nu

Introdus: HTML 3.2

Browsere: Netscape Navigator 2.0, 3.x, 4.x; Microsoft Internet Explorer 2.0, 3.0, 4.0.

Exemplu:
```
<P><BIG>HEY</BIG>, come over here!
```

<I>...</I>

Utilizare: Este folosit pentru a scrie textul cu caractere italice.

Sintaxa: `<I [atribute]>textul scris cu caractere italice</I>`

Eticheta de start/stop: Cerută /Cerută

Atribute: `class=, id=, style=, title=, lang=, dir=` Atribute uzuale.

Intrinseci: `onclick=, ondblclick=, onmousedown=, onmouseup=, onmouseover=, onmousemove=, onmouseout=, onkeypress=, onkeydown=, onkeyup=` Evenimente intrinseci uzuale.

Gol: Nu

Introdus: HTML 2.0

Browsere: Netscape Navigator 1.0, 2.x, 3.x, 4.x; Microsoft Internet Explorer 2.0, 3.0, 4.0.

Exemplu:
```
<P><I>HTML 4 Unleashed, Professional Reference Edition</I>
      is available from Sams.net Publishing.
```

<S>...</S>

Utilizare: Este folosit pentru a scrie textul tăiat cu linie orizontală.

Sintaxa: `<S [atribute]>textul taiat cu linie orizontala</S>`

Eticheta de start/stop: Cerută /Cerută

Atribute: `class=, id=, style=, title=, lang=, dir=` Atribute uzuale.

Intrinseci: `onclick=, ondblclick=, onmousedown=, onmouseup=, onmouseover=, onmousemove=, onmouseout=, onkeypress=, onkeydown=, onkeyup=` Evenimente intrinseci uzuale.

Gol: Nu

Observații: Este nerecomandat. Folosiți eticheta DEL dacă este posibil.

Observații: Este similar cu elementul STRIKE.

Introdus: HTML 3.2

Browsere: Netscape Navigator 3.0, 4.0; Microsoft Internet Explorer 3.0, 4.0.

Exemplu:
```
<P><S>Contractul expira în 7 octombrie 1997</S>
```

<SMALL>...</SMALL>

Utilizare: Este folosit pentru a scrie textul micșorat.

Sintaxa: `<SMALL[atribute]>text micsorat</SMALL>`

Eticheta de start/stop: Cerută /Cerută

Atribute: `class=, id=, style=, title=, lang=, dir=` Atribute uzuale.

Intrinseci: `onclick=, ondblclick=, onmousedown=, onmouseup=, onmouseover=, onmousemove=, onmouseout=, onkeypress=, onkeydown=, onkeyup=` Evenimente intrinseci uzuale.

Gol: Nu

Observații: Aceleași operații sunt acum realizate de foile de stil.

Introdus: HTML 3.2

Browsere: Netscape Navigator 2.0, 3.0, 4.0; Microsoft Internet Explorer 2.0, 3.0, 4.0.

Exemplu:
```
<P><SMALL>This is fine print.</SMALL>
```

<STRIKE>...</STRIKE>

Utilizare: Este folosit pentru a scrie textul tăiat cu o linie.

Sintaxa: `<STRIKE [atribute]>textul taiat cu o linie</STRIKE>`

Eticheta de start/stop: Cerută /Cerută

Atribute: `class=, id=, style=, title=, lang=, dir=` Atribute uzuale.

Intrinseci: `onclick=, ondblclick=, onmousedown=, onmouseup=, onmouseover=, onmousemove=, onmouseout=, onkeypress=, onkeydown=, onkeyup=` Evenimente intrinseci uzuale.

Gol: Nu

Observații: Este nerecomandat. Folosiți eticheta DEL dacă este posibil.

Introdus: HTML 3.2

Browsere: Netscape Navigator 3.0, 4.0; Microsoft Internet Explorer 2.0, 3.0, 4.0.

Exemplu:
```
<P><STRIKE> Contractul expira în 7 octombrie 1997</STRIKE>
```

<TT>...</TT>

Utilizare: Este folosit pentru a scrie textul în mod teletype (monospațiat).

Sintaxa: `<TT [atribute]>textul tip teletype</TT>`

Eticheta de start/stop: Cerută /Cerută

Atribute: `class=, id=, style=, title=, lang=, dir=` Atribute uzuale.

Intrinseci: `onclick=, ondblclick=, onmousedown=, onmouseup=, onmouseover=, onmousemove=, onmouseout=, onkeypress=, onkeydown=, onkeyup=` Evenimente intrinseci uzuale.

Gol: Nu

Observații: Este considerat adesea ca un text de tipul „mașină de scris".

Introdus: HTML 3.2

Browsere: Netscape Navigator 1.0, 2.0, 3.x, 4.x; Microsoft Internet Explorer 2.0, 3.0, 4.0.

Exemplu:

```
<P>The<TT>Teletype element</TT> can be used to represent many
➥ things like code, directory listings and other
➥ elements that need to stand out.
```

<U>...</U>[0]

Utilizare: Este folosit pentru a scrie textul subliniat.

Sintaxa: `<U [atribute]>text subliniat</U>`

Eticheta de start/stop: Cerută /Cerută

Atribute: `class=, id=, style=, title=, lang=, dir=` Atribute uzuale.

Intrinseci: `onclick=, ondblclick=, onmousedown=, onmouseup=, onmouseover=, onmousemove=, onmouseout=, onkeypress=, onkeydown=, onkeyup=` Evenimente intrinseci uzuale.

Gol: Nu

Observații: Este nerecomandat.

Introdus: HTML 3.2

Browsere: Netscape Navigator 3.0, 4.0; Microsoft Internet Explorer 2.0, 3.0, 4.0.

Exemplu:

```
<P><U> HTML 4 Unleashed, Professional Reference Edition</U>
      is available from Sams.net Publishing.
```

Marcatoare de expresie: ACRONYM, ADDRESS*, BLOCKQUOTE*, CITE, CODE, DEL, DFN, EM, INS, KBD, PRE*, SAMP, STRONG, VAR

Marcatoarele de expresie pun accentul pe înțelesul lexical al unei expresii sau al unui cuvânt, identificându-le după locul lor în structura propoziției și nu după o caracteristică fizică a fontului. Această metodă de identificare facilitează căutarea lor în utilizarea metodelor automate de căutare și simplifică crearea echivalențelor în limbi străine.

<ACRONYM>...</ACRONYM>

Utilizare: Este folosit pentru a defini acronimele.

Sintaxa: `<ACRONYM [atribute]>textul cu acronime</ACRONYM>`

Eticheta de start/stop: Cerută /Cerută

Atribute: `class=, id=, style=, title=, lang=, dir=` Atribute uzuale.

Intrinseci: `onclick=, ondblclick=, onmousedown=, onmouseup=, onmouseover=, onmousemove=, onmouseout=, onkeypress=, onkeydown=, onkeyup=` Evenimente intrinseci uzuale.

Gol: Nu

Observații: Folosiți atributul `title` pentru a expanda acronimul. Această etichetă este importantă în special pentru browserele audio deoarece îi spune browserului să spună acronimul literă cu literă în loc să îl pronunțe, „Eff Bee Eye" este mai ușor de auzit și de înțeles în loc de „phbee".

Introdus: HTML 4.0

Browsere: Netscape Navigator 4.0; Microsoft Internet Explorer 3.0.

Exemplu:

```
The<ACRONYM title="Federal Bureau of Investigation">FBI
➥</ACRONYM>
```

<ADDRESS>...</ADDRESS>*

Observații: Operează ca un element la nivel de bloc; a fost prezentat în secțiunea „Elemente la nivel de bloc", ca fiind un element de bloc de bază.

<BLOCKQUOTE>...</BLOCKQUOTE>*

Observații: Operează ca un element la nivel de bloc; a fost prezentat la capitolul „Elemente la nivel de bloc", ca fiind un element de bloc de bază.

<CITE>...</CITE>

Utilizare: Citează o referință.

Sintaxa: `<CITE [atribute]>continutul citatului</CITE>`

Eticheta de start/stop: Cerută /cerută

Atribute: `class=, id=, style=, title=, lang=, dir=` Atribute uzuale.

Intrinseci: `onclick=, ondblclick=, onmousedown=, onmouseup=, onmouseover=, onmousemove=, onmouseout=, onkeypress=, onkeydown=, onkeyup=` Evenimente intrinseci uzuale.

Gol: Nu

Introdus: HTML 2.0

Browsere: Netscape Navigator 1.0, 2.0, 3.x, 4.x; Microsoft Internet Explorer 2.0, 3.0, 4.0.

Exemplu:
```
<P>
She drinks the honey of her vision. <CITE>(Mira Bai, c.
➥1530)</CITE>
```

<CODE>...</CODE>

Utilizare: Identifică o linie sau o secțiune de cod pentru afișare.

Sintaxa: `<CODE [atribute]>textul în format cod</CODE>`

Eticheta de start/stop: Cerută /Cerută

Atribute: `class=, id=, style=, title=, lang=, dir=` Atribute uzuale.

Intrinseci: `onclick=, ondblclick=, onmousedown=, onmouseup=, onmouseover=, onmousemove=, onmouseout=, onkeypress=, onkeydown=, onkeyup=` Evenimente intrinseci uzuale.

Gol: Nu

Introdus: HTML 2.0

Browsere: Netscape Navigator 1.0, 2.0, 3.x, 4.x; Microsoft Internet Explorer 2.0, 3.0, 4.0.

Exemplu:
```
<P>Trebuie sa va salvati fisierele in directorul <CODE>
➥c:\examples </CODE>.
```

<DEL>...</DEL>

Utilizare: Arată că textul a fost șters din document la ultima modificare.

Sintaxa: `<DEL [atribute]>textul sters</DEL>`

Eticheta de start/stop: Cerută /Cerută

Atribute: `class=, id=, style=, title=, lang=, dir=` Atribute uzuale.

Intrinseci: `onclick=, ondblclick=, onmousedown=, onmouseup=, onmouseover=, onmousemove=, onmouseout=, onkeypress=, onkeydown=, onkeyup=` Evenimente intrinseci uzuale.

`cite="..."` Este adresa URL a documentului sursă.

`datetime="..."` Indică data și ora schimbării.

Gol: Nu

Observații: Este un element nou în HTML 4.0

Introdus: HTML 4.0

Browsere: Netscape Navigator 4.0; Microsoft Internet Explorer 4.0.

Exemplu:
```
<HTML>
  <BODY>
    <P>Type in <DEL cite="revision.html"
         datetime="1997-10-31T23:15:30 - 08:00">Enter</DEL>
         <KBD>red0202</KBD>
       as your password.
  </BODY>
</HTML>
```


<DFN>...</DFN>

Utilizare: Definește un termen inclus.

Sintaxa: `<DFN [atribute]>continutul definitiei</DFN>`

Eticheta de start/stop: Cerută /Cerută

Atribute: `class=, id=, style=, title=, lang=, dir=` Atribute uzuale.

Intrinseci: `onclick=, ondblclick=, onmousedown=, onmouseup=, onmouseover=, onmousemove=, onmouseout=, onkeypress=, onkeydown=, onkeyup=` Evenimente intrinseci uzuale.

Gol: Nu

Introdus: HTML 2.0

Browsere: Netscape Navigator 3.x, 4.x ; Microsoft Internet Explorer 2.0, 3.0, 4.0.

Exemplu:
```
<DFN>Paznic: Este un angajat care are responsabilitatea de a
     ➥controla accesul persoanelor la o cale de acces sau
     ➥intr-o zona</DFN>
```

<EM>...</EM>

Utilizare: Este folosit pentru a evidenția textul.

Sintaxa: `<EM [atribute]>textul evidentiat</EM>`

Eticheta de start/stop: Cerută /Cerută

Atribute: `class=, id=, style=, title=, lang=, dir=` Atribute uzuale.

Intrinseci: `onclick=, ondblclick=, onmousedown=, onmouseup=, onmouseover=, onmousemove=, onmouseout=, onkeypress=, onkeydown=, onkeyup=` Evenimente intrinseci uzuale.

Gol: Nu

Introdus: HTML 3.2

Browsere: Netscape Navigator 1.0, 2.0, 3.x, 4.x; Microsoft Internet Explorer 2.0, 3.0, 4.0.

Exemplu:
```
<P>Acest cuvânt este <EM>evidentiat!</EM>
```

<INS>...</INS>

Utilizare: Arată că textul a fost inserat în document la ultima modificare.

Sintaxa: `<INS [atribute]>textul inserat</INS>`

Eticheta de start/stop: Cerută /Cerută

Atribute: `cite="..."` Este adresa URL a documentului sursă.

`datetime="..."` Indică data și ora modificării.

`class=, id=, style=, title=, lang=, dir=` Atribute uzuale.

Intrinseci: `onclick=, ondblclick=, onmousedown=, onmouseup=, onmouseover=, onmousemove=, onmouseout=, onkeypress=, onkeydown=, onkeyup=` Evenimente intrinseci uzuale.

Gol: Nu

Observații: Este un element nou în limbajul HTML 4.0.

Introdus: HTML 4.0

Browsere: Netscape Navigator 4.0; Microsoft Internet Explorer 4.0.

Exemplu:
```
<INS cite ="revision.html"
     datetime="1997-10-31T23:15:30-08:00">Enter</INS>
     ➥<KBD>red0202</KBD> as your password.
```

<KBD>...</KBD>

Utilizare: Indică un text pe care un utilizator ar trebui să-l introducă de la tastatură.

Sintaxa: `<KBD [atribute]>textul care trebuie introdus</KBD>`

Eticheta de start/stop: Cerută /Cerută

Atribute: `class=, id=, style=, title=, lang=, dir=` Atribute uzuale.

Intrinseci: `onclick=, ondblclick=, onmousedown=, onmouseup=, onmouseover=, onmousemove=, onmouseout=, onkeypress=, onkeydown=, onkeyup=` Evenimente intrinseci uzuale.

Gol: Nu

Introdus: HTML 2.0

Browsere: Netscape Navigator 1.0, 2.0, 3.x, 4.x; Microsoft Internet Explorer 2.0, 3.0, 4.0.

Exemplu:
```
<P>Type in <KBD>red0202</KBD> as your password.
```

<PRE>...</PRE>*

Observații: Operează ca un element la nivel de bloc; a fost prezentat la capitolul „Elemente la nivel de bloc", ca fiind un element de bloc de bază.

<SAMP>...</SAMP>

Utilizare: Identifică un exemplu de ieșire așa cum ar apărea la ecranul calculatorului sau la imprimantă.

Sintaxa: `<SAMP [atribute]>exemplu de iesire</KBD>`

Eticheta de start/stop: Cerută /Cerută

Atribute: `class=, id=, style=, title=, lang=, dir=` Atribute uzuale.

Intrinseci: `onclick=, ondblclick=, onmousedown=, onmouseup=, onmouseover=, onmousemove=, onmouseout=, onkeypress=, onkeydown=, onkeyup=` Evenimente intrinseci uzuale.

Gol: Nu

Introdus: HTML 3.2

Browsere: Netscape Navigator 1.0, 2.0, 3.x, 4.x; Microsoft Internet Explorer 2.0, 3.0, 4.0.

Exemplu:
```
<P>The ticket printer will eject the completed sales receipt
   ➥containing the date and amount of purchase:<BR><BR>
<SAMP>Oct 31, 1997<BR>
      Purchase Amount: $32.56<BR>
      Thank you for shopping at XYZ Market
</SAMP>
```

<STRONG>...</STRONG>

Utilizare: Permite o evidențiere mai puternică a textului.

Sintaxa: `<STRONG [atribute]>textul cu evidentiere mai puternica</STRONG>`

Eticheta de start/stop: Cerută /Cerută.

Atribute: `class=, id=, style=, title=, lang=, dir=` Atribute uzuale.

Intrinseci: `onclick=, ondblclick=, onmousedown=, onmouseup=, onmouseover=, onmousemove=, onmouseout=, onkeypress=, onkeydown=, onkeyup=` Evenimente intrinseci uzuale.

Gol: Nu

Introdus: HTML 2.0

Browsere: Netscape Navigator 1.0, 2.0, 3.x, 4.x; Microsoft Internet Explorer 2.0, 3.0, 4.0.

Exemplu:
```
...
<P><STRONG>Atentie:</STRONG> Luati masuri de precautie
inainte de a aprinde focul.
...
```

<VAR>...</VAR>

Utilizare: Definește o variabilă.

Sintaxa: `<VAR [atribute]>variabila</VAR>`

Eticheta de start/stop: Cerută/Cerută

Atribute: `class=, id=, style=, title=, lang=, dir=` Atribute uzuale.

Intrinseci: `onclick=, ondblclick=, onmousedown=, onmouseup=, onmouseover=, onmousemove=, onmouseout=, onkeypress=, onkeydown=, onkeyup=` Evenimente intrinseci uzuale.

Gol: Nu

Introdus: HTML 2.0

Browsere: Netscape Navigator 2.x, 3.x, 4.x; Microsoft Internet Explorer 2.x, 3.x, 4.x.

Exemplu:
```
...
<P><VAR>Numele serverului virtual</VAR> trebuie sa fie
   ➥identic cu numele serverului dumneavoastra de Web.
...
```

Marcatoare speciale: A, APPLET°, BASEFONT°, BDO, BR, FONT°, IFRAME*, IMG, NOSCRIPT*, OBJECT, Q, SCRIPT*, SPAN, SUB, SUP

<A>...</A>

Utilizare: Este folosit pentru a defini legăturile și ancorele.

Sintaxa: `<A [atribute]>textul ancorei</A>`

Eticheta de start/stop: Cerută /Cerută

Atribute:

`charset="..."` codificarea caracterelor resursei. valoarea prestabilită este `ISO-8859-1`.

`name="..."` Definește o ancoră.

`href="..."` Este adresa URL a resursei legate.

`target="..."` Determină unde va fi afișată resursa (nume dat de utilizator, `_blank`, `_parent`, `_self`, `_top`).

`rel="..."` Tipurile de legături forward (înainte).

`rev="..."` Tipurile de legături reverse (înapoi).

`accesskey="..."` Atribuie o tastă rapidă acestui element.

`shape="..."` Vă permite să definiți hărți imagine pe partea clientului folosind modele definite (`default` (prestabilit), `rect` (dreptunghi), `circle` (cerc), `poly` (poligon)).

`coords="..."` Stabilește dimensiunea modelului în lungimi date în pixeli sau procente.

`tabindex="..."` Stabilește ordinea de tabulare între elemente, cu un index de tabulare definit.

`class=, id=, style=, title=, lang=, dir=` Atribute uzuale.

Intrinseci: `onclick=, ondblclick=, onmousedown=, onmouseup=, onmouseover=, onmousemove=, onmouseout=, onkeypress=, onkeydown=, onkeyup=` Evenimente intrinseci uzuale.

Gol: Nu

Introdus: HTML 1.0

Browsere: Netscape Navigator 1.0, 1.1, 2.x, 3.x, 4.x; Microsoft Internet Explorer 2.0, 3.0, 4.0.

Exemplu:

```
<HTML>
  <HEAD>
    <TITLE>Link example</TITLE>
  </HEAD>
  <BODY>
    <P>Trimitere la: <A href="example.htm">Exemplu</A>
  </BODY>
</HTML>
```


<APPLET>...</APPLET>°

Utilizare: Conține o miniaplicație scrisă în limbajul Java.

Sintaxa: `<APPLET [atribute]>continutul miniaplicatiei</APPLET>`

Eticheta de start/stop: Cerută /Cerută

Atribute:

`codebase="..."` Este adresa URL de bază pentru miniaplicație.

`archive="..."` Identifică resursele care trebuie preîncărcate.

`code="..."` Reprezintă fișierul clasă al miniaplicației.

`object="..."` Reprezintă fișierul serializat al miniaplicației.

`alt="..."` Afișează un text în timpul încărcării.

`name="..."` Definește numele miniaplicației.

`width="..."` Definește lățimea miniaplicației afișate.

`height="..."` Definește înălțimea miniaplicației afișate.

`align="..."` Este nerecomandat. Controlează alinierea (`left`, `center`, `right`, `justify`)

`hspace="..."` Definește spațiul orizontal care separă imaginea de alt conținut.

`vspace="..."` Definește spațiul vertical care separă imaginea de alt conținut.

Gol: Nu

Observații: Miniaplicația este nerecomandată, fiind folosit în locul ei, elementul `OBJECT`. A fost inițial una din extensiile Netscape introduse pentru a accepta miniaplicațiile încorporate în Java pentru paginile limbajului HTML.

Introdus: HTML 3.2

Browsere: Netscape Navigator 2.0, 3.x, 4.x; Microsoft Internet Explorer 2.0, 3.0, 4.0.

Exemplu:

```
<APPLET code ="name.class" width="pixels" height="pixels">
</APPLET>
```

<BASEFONT>°

Utilizare: Stabilește dimensiunea fontului de bază.

Sintaxa: `<BASEFONT [atribute]>`

Eticheta de start/stop: Cerută /Interzisă

Atribute:

`size="..."` Definește dimensiunea fontului (1 până la 7 sau relativă, cum ar fi de exemplu +3).

`color="..."` Definește culoarea fontului.

`face="..."` Definește tipul fontului.

Gol: Da, eticheta de încheiere este interzisă.

Observații: Este nerecomandat, fiind folosite în schimb foile de stil pentru că folosirea acestui element tinde să cauzeze probleme uriașe utilizatorilor care nu au fonturi ne-europene și utilizatorilor care depind de anumite caracteristici ale fonturilor.

Introdus: HTML 3.2

Browsere: Netscape Navigator 1.0, 2.0, 3.x, 4.x; Microsoft Internet Explorer 2.0, 3.0, 4.0.

Exemplu :

```
<P><BASEFONT size="2"><B>Numele serverului virtual</B>
➥trebuie sa fie identic cu numele serverului dumneavoastra
➥de Web.
```

<BDO>...</BDO>

Utilizare: Elementul BDO (impunerea bidirecțională) este folosit pentru a modifica selectiv direcția prestabilită a textului inline.

Sintaxa: `<BDO [atribute]>continutul BDO</BDO>`

Eticheta de start/stop: Cerută /Cerută

Atribute: `lang="..."` Definește limba documentului.

`dir="..."` Definește direcția textului (ltr, rtl)

Gol: Nu

Introdus: HTML 4.0

Browsere: Nici unul la ora actuală.

Observații: Atributul `dir` este obligatoriu. Elementul BDO a fost propus pentru a fi integrat în noua versiune, pentru intenaționalizarea limbajului HTML. Este necesar pentru limbi cum ar fi araba sau ebraica, în care literele se scriu de la dreapta la stânga dar pot avea elemente inline, cum ar fi includerea cuvintelor sau numerelor englezești, care ar trebui să se citească de la stânga la dreapta. Această etichetă poate fi de asemenea folosită pentru a include cuvinte individuale în orice limbă, cu o ordine de scriere diferită, pentru a crea un efect special.

Exemplu:

```
... <BDO dir="rtl">.rorrim a ni gnikool ton er'uoy , siht
➥daer nac uoy fI</BDO> ...
```


Utilizare: Forțează o întrerupere de rând.

Sintaxa: `<BR [atribute]>`

Eticheta de start/stop: Cerută /Interzisă

Atribute: `clear="..."` Stabilește locul de unde începe următoarea linie după un obiect flotant (`none`, **`left`**, `right`, `all`)

`class=, id=, style=, title=, lang=, dir=` Atribute uzuale.

Intrinseci: `onclick=, ondblclick=, onmousedown=, onmouseup=, onmouseover=, onmousemove=, onmouseout=, onkeypress=, onkeydown=, onkeyup=` Evenimente intrinseci uzuale.

Gol: Da, eticheta de încheiere este interzisă.

Introdus: HTML 1.0

Browsere: Netscape Navigator 1.0, 2.0, 3.x, 4.x; Microsoft Internet Explorer 2.0, 3.0, 4.0.

Exemplu:

```
...
<P> Eticheta &lt; BR&gt. este folosita pentru a determina o
    intrerupere de rand intre exemple.<BR>
    Eticheta &lt;BR&gt; este folosita pentru a determina o
    intrerupere de rand între exemple.<BR>
    Eticheta &lt;BR&gt; este introdusa in acest fragment de
    cod pentru a fi afisata corect pe pagina, fiind folosita
    si pentru a controla intreruperile de rand. Aceasta
    tehnica este utila deseori la crearea paginilor tip
    tutorial care trebuie sa afiseze cod HTML fara a-l
    interpreta.
...
```

<FONT>...</FONT>[0]

Utilizare: Modifică dimensiunea și culoarea fontului.

Sintaxa: `<FONT [atribute]>textul</FONT>`

Eticheta de start/stop: Cerută /Cerută

Atribute: `size="..."` Definește dimesiunea fontului (1 până la 7 sau relativă, cum ar fi de exemplu +3).

`color="..."` Definește culoarea fontului.

`face="..."` Definește tipul fontului.

Gol: Nu

Observații: Este nerecomandat, fiind folosite în schimb foile de stil pentru că folosirea acestui element tinde să cauzeze probleme uriașe utilizatorilor care nu au fonturi ne-europene și utilizatorilor care depind de anumite caracteristici ale fonturilor.

Introdus: HTML 3.2

Browsere: Netscape Navigator 1.1, 2.0, 3.x, 4.x; Microsoft Internet Explorer 2.0, 3.0, 4.0.

Exemplu:

```
<P><Font face="Arial">Acesta este un exemplu de utilizare a
      elementului FONT pentru a defini tipul ARIAL</FONT>
```

<IFRAME>...</IFRAME>*

Observații: Operează ca un element inline în interiorul corpului documentului, dar a fost mai comod prezentat ca un element pentru cadre la capitolul „Elemente structurale (de organizare)", unde de fapt nu este permis.

<IMG>

Utilizare: Include o imagine într-un document.

Sintaxa: `<IMG [atribute]>`

Eticheta de start/stop: Cerută /Interzisă

Atribute:

`src="..."` Reprezintă adresa URL a imaginii.

`alt="..."` Definește un text alternativ pentru afișare.

`align="..."` Este nerecomandat. Controlează alinierea (**`bottom`**, `middle`, `top`, `left`, `right`).

`height="..."` Definește înălțimea imaginii.

`width="..."` Definește lățimea imaginii.

`border="..."` Definește lățimea bordurii.

`hspace="..."` Definește spațiul orizontal care separă imaginea de alt conținut.

`vspace="..."` Definește spațiul vertical care separă imaginea de alt conținut.

`usemap="..."` Reprezintă adresa URL a unei hărți de imagini pe partea clientului.

`ismap` Identifică o hartă de imagini pe partea serverului.

`class=, id=, style=, title=, lang=, dir=` Atribute uzuale.

Intrinseci: `onclick=, ondblclick=, onmousedown=, onmouseup=, onmouseover=, onmousemove=, onmouseout=, onkeypress=, onkeydown=, onkeyup=` Evenimente intrinseci uzuale.

Gol: Da, eticheta de încheiere este interzisă.

Introdus: HTML 2.0

Browsere: Netscape Navigator 1.0, 2.x, 3.x, 4.x; Microsoft Internet Explorer 2.0, 3.0, 4.0.

Exemplu:

```
<HTML>
    <HEAD>
      <TITLE>Picture</TITLE>
    </HEAD>
    <BODY>
      <Img src="picture1.jpg" alt="[Autoportret]">
    </BODY>
</HTML>
```

<NOSCRIPT>...</NOSCRIPT>°

Observații: Operează ca un element inline în interiorul corpului documentului, dar a fost mai comod prezentat la capitolul „Elemente la nivel de bloc", ca fiind un element de bloc de bază, unde de fapt nu este permis.

<OBJECT>...</OBJECT>

Utilizare: Inserează un obiect.

Sintaxa: `<OBJECT [atribute]>continutul obiectului</OBJECT>`

Eticheta de start/stop: Cerută /Cerută

Atribute:

`declare` Este un indicator care declară, dar nu creează un obiect.

`classid="..."` Reprezintă adresa URL a locației obiectului.

`codebase="..."` Reprezintă adresa URL pentru conversia adreselor URL specificate de alte atribute.

`data="..."` Reprezintă adresa URL a datelor obiectului.

`type="..."` Definește tipul conținutului Internet pentru date.

`codetype="..."` Definește tipul conținutului Internet pentru cod.

`standby="..."` Afișează un mesaj în timp ce se face încărcarea.

`align="..."` Este nerecomandat. Controlează alinierea (`texttop` (partea de sus a textului), `middle` (mijloc), `textmiddle` (partea de mijloc a textului), `baseline` (linia de bază), `textbottom` (partea de jos a textului), `left` , `center` , `right` , `justify` (aliniere la stânga și la dreapta)).

`height="..."` Definește înălțimea obiectului.

`width="..."` Definește lățimea obiectului.

`border="..."` Afișează bordura în jurul obiectului.

`hspace="..."` Definește spațiul dintre marginile obiectului și alt conținut al paginii.

`vspace="..."` Definește spațiul dintre partea de sus și partea de jos a obiectului și alt conținut al paginii.

`usemap="..."` Reprezintă adresa URL spre o hartă de imagini.

`shapes="..."` Vă permite să definiți zone de căutare a hiperlegăturilor, dacă obiectul este o imagine.

`name="..."` Reprezintă adresa URL unde puteți să trimiteți obiectul ca parte dintr-un formular.

`tabindex="..."` Stabilește ordinea de tabulare între elemente folosind un index de tabulare definit.

`class=, id=, style=, title=, lang=, dir=` Atribute uzuale.

Intrinseci: `onclick=, ondblclick=, onmousedown=, onmouseup=, onmouseover=, onmousemove=, onmouseout=, onkeypress=, onkeydown=, onkeyup=` Evenimente intrinseci uzuale.

Gol: Nu

Observații: Acest element poate fi imbricat la orice nivel cu nivelul cel mai din exterior la care browserul este capabil să oprească recursiunea. Nivelul cel mai din interior trebuie să fie de tip text pentru a

permite unui număr cât mai mare de browsere să asigure un conținut care să aibă sens.

Introdus: HTML 4.0

Browsere: Netscape Navigator 3.x, 4.x; Microsoft Internet Explorer 2.0, 3.0, 4.0.

Exemplu:

```
...
<Object title="Barking dog"
  classid="Barking.py">
  <OBJECT data="Barking.mpeg" type="application/mpeg">
    <OBJECT src="Barking.gif">
     [Barking Dog Logo]
    </OBJECT>
  </OBJECT>
</OBJECT>
...
```

<PARAM>

Utilizare: Inițializează un obiect.

Sintaxa: `<PARAM [atribute]>`

Eticheta de start/stop: Cerută /Interzisă.

Atribute:

`name="..."` Definește numele parametrului.

`value="..."` Definește valoarea parametrului obiectului.

`valuetype="..."` Definește tipul valorii (**data** (dată), `ref` (referință), `object` (obiect))

`type="..."` Definește tipul mediului Internet.

Gol: Da, eticheta de încheiere este interzisă.

Observații: Trebuie să preceadă alt conținut de obiect. Este folosit cu etichetele nerecomandate `<OBJECT>` și `<APPLET>`.

Introdus: HTML 3.2

Browsere: Netscape Navigator 1.0, 2.0, 3.x, 4.x; Microsoft Internet Explorer 2.0, 3.0, 4.0.

Exemplu:

```
<HTML>
  <HEAD>
  </HEAD>
  <BODY>
    ...
    <APPLET code="tickertape.class" width="400" height="40">
     <PARAM name="message" value="This is a test message.">
     <PARAM name="speed" value="10">
     <PARAM name="textcolor" value="120, 120,0">
    </APPLET>
    ...
  </BODY>
</HTML>
```


<Q>...</Q>

Utilizare: Folosit pentru a afișa citatele scurte care nu necesită o întrerupere de paragraf.

Sintaxa: `<Q [atribute]>citat scurt</Q>`

Eticheta de start/stop: Cerută /Cerută

Atribute:

`cite="..."` Adresa URL a textului citat.

`class=, id=, style=, title=, lang=, dir=` Atribute uzuale.

Intrinseci: `onclick=, ondblclick=, onmousedown=, onmouseup=, onmouseover=, onmousemove=, onmouseout=, onkeypress=, onkeydown=, onkeyup=` Evenimente intrinseci uzuale.

Gol: Nu

Observații: Unele browsere încadrează cu ghilimelele corespunzătoare limbii citatul inline. Este un element nou în limbajul HTML 4.0.

Introdus: HTML 4.0

Browsere: Netscape Navigator 4.0; Microsoft Internet Explorer 4.0.

Exemplu:

```
<Q cite="http://www.enheduanna.com">O lady of all truths...
➥</Q>
```

<SCRIPT>...</SCRIPT>*

Observații: Operează ca un element inline în interiorul corpului documentului, dar a fost discutat la capitolul „Elemente structurale (de organizare)" ca un element de titlu, unde este de asemenea permis.

<SPAN>...</SPAN>

Utilizare: Organizează documentul, prin definirea unui interval de text inline.

Sintaxa: `<SPAN [atribute]>Intervalul definit pentru text</SPAN>`

Eticheta de start/stop: Cerută /Cerută

Atribute: `class=, id=, style=, title=, lang=, dir=` Atribute uzuale.

Intrinseci: `onclick=, ondblclick=, onmousedown=, onmouseup=, onmouseover=, onmousemove=, onmouseout=, onkeypress=, onkeydown=, onkeyup=` Evenimente intrinseci uzuale.

Gol: Nu

Observații: Este echivalent cu elementul `<DIV>` dar operează ca un element de text inline, în loc să forțeze o întrerupere.

Introdus: HTML 3.2

Browsere: Netscape Navigator 3.0, 4.x; Microsoft Internet Explorer 2.0, 3.0, 4.0.

Exemplu:

```
<P>
<DIV id="Books-HTML" class="Books">
  Book title: <Span class="Books-Title">HTML 4 Unleashed,
```

```
    Proffesional Reference Edition</SPAN>
    ISBN:<SPAN class="Books-ISBN">1-57521-380-X</SPAN>
  </DIV>
```

_{...}

Utilizare: Creează un indice.

Sintaxa: `<SUB [atribute]>textul indicelui</SUB>`

Eticheta de start/stop: Cerută /Cerută

Atribute: `class=, id=, style=, title=, lang=, dir=` Atribute uzuale.

Intrinseci: `onclick=, ondblclick=, onmousedown=, onmouseup=, onmouseover=, onmousemove=, onmouseout=, onkeypress=, onkeydown=, onkeyup=` Evenimente intrinseci uzuale.

Gol: Nu

Observații: Este folosit în notațiile științifice și în unele limbi.

Introdus: HTML 3.2

Browsere: Netscape Navigator 2.0, 3.x, 4.x; Microsoft Internet Explorer 2.0, 3.0, 4.0.

Exemplu:

```
<P>Formula chimica a apei este: H<SUB>2</SUB>O
```

^{...}

Utilizare: Creează un exponent.

Sintaxa: `<SUP [atribute]>textul exponentului</SUP>`

Eticheta de start/stop: Cerută/Cerută

Atribute: `class=, id=, style=, title=, lang=, dir=` Atribute uzuale.

Intrinseci: `onclick=, ondblclick=, onmousedown=, onmouseup=, onmouseover=, onmousemove=, onmouseout=, onkeypress=, onkeydown=, onkeyup=` Evenimente intrinseci uzuale.

Gol: Nu

Observații: Este folosit la abrevierile scrise cu exponent, în anumite limbi și în notații cu exponent.

Introdus: HTML 3.2

Browsere: Netscape Navigator 2.0, 3.x, 4.x; Microsoft Internet Explorer 2.x, 3.x, 4.x.

Exemplu:

```
<P>M<SUP>lle</SUP> Robards and W<SUP>M</SUP> Shakespeare
met on 3<SUP>rd</SUP> Street
<P>Aceasta este o egalitate simpla: 3<SUP>3</SUP> = 27
```


Hărți imagine pe partea clientului: AREA, MAP

<AREA>

Utilizare: Elementul AREA este folosit pentru a defini legături și ancore.

Sintaxa: `<AREA [atribute]>`

Eticheta de start/stop: Cerută /Interzisă

Atribute:

`shape="..."` Vă permite să definiți hărți imagine pe partea clientului, prin folosirea mdodelelor definite (`default` (prestabilit), `rect` (dreptunghi), `circle` (cerc), `poly` (poligon)).

`coords=".."` Stabilește dimensiunea modelului folosind unități de lungime date în pixeli sau procente.

`href="..."` Reprezintă adresa URL a resursei legate.

`target="..."` Determină unde va fi afișată resursa. (nume definit de utilizator, `_blank`, `_parent`, `_self`, `_top`).

`nohref="..."` Indică faptul că zona respectivă nu conține nici o acțiune.

`alt="..."` Afișează un text alternativ.

`tabindex="..."` Stabilește ordinea de tabulare între elemente cu un index de tabulare definit.

Gol: Da, eticheta de încheiere este interzisă.

Observații: Este valabil doar în cadrul unui container MAP.

Introdus: HTML 3.2

Browsere: Netscape Navigator 1.0, 2.0, 3.x, 4.x; Microsoft Internet Explorer 2.0, 3.0, 4.0.

Exemplu:

```
<HTML>
<MAP name="page1">
  <AREA shape="circle" coords="70,70,35">
  <AREA shape="rectangle" coords="65, 65, 80, 80">
</MAP>
```

<MAP>...</MAP>

Utilizare: Când este folosit cu un element AREA, creează o hartă imagine pe partea clientului.

Sintaxa: `<MAP [atribute]>continutul hartii imagine</MAP>`

Eticheta de start/stop: Cerută /Cerută

Atribute: `name="..."` Definește numele hărții imagine care trebuie creată.

`class=, id=, style=, title=, lang=, dir=` Atribute uzuale.

Gol: Nu

Introdus: HTML 3.2

Browsere: Netscape Navigator 1.0, 2.x, 3.x, 4.x; Microsoft Internet Explorer 2.0, 3.0, 4.0.

Exemplu:

```
<HTML>
<MAP name="page1">
<AREA shape="circle"coords="70,70,35">
<AREA shape="rectangle"coords="65, 65, 80, 80">
</MAP>
```

Text pentru controlul formularelor: BUTTON*, INPUT*, LABEL*, SELECT*, TEXTAREA*

<BUTTON>...</BUTTON>*

Observații: Operează ca un element inline în interiorul corpului unui formular; a fost discutat la secțiunea „Elemente la nivel de bloc", ca fiind un element de formular, unde este permis.

<INPUT>...</INPUT>*

Observații: Operează ca un element inline în interiorul corpului unui formular; a fost discutat la secțiunea „Elemente la nivel de bloc", ca fiind un element de formular, unde este permis.

<LABEL>...</LABEL>*

Observații: Operează ca un element inline în interiorul corpului unui formular; a fost discutat la secțiunea „Elemente la nivel de bloc", ca fiind un element de formular, unde este permis.

<SELECT>...</SELECT>*

Observații: Operează ca un element inline în interiorul corpului unui formular; a fost discutat la secțiunea „Elemente la nivel de bloc", ca fiind un element de formular, unde este permis.

<TEXTAREA>...</TEXTAREA>*

Observații: Operează ca un element inline în interiorul corpului unui formular; a fost discutat la secțiunea „Elemente la nivel de bloc", ca fiind un element de formular, unde este permis.

Atribute uzuale și evenimente intrinseci

Pentru majoritatea elementelor se aplică patru atribute de bază:

- `id="..."` Este un identificator global unic, care permite ca elementul să fie distribuit individual de către o foaie de stil sau alt proces.
- `class="..."` Este o listă de clase, cu identificatori multipli, separați prin spații libere, care permite ca elementul să fie distribuit ca un grup, de către o foaie de stil sau alt proces.
- `style="..."` Reprezintă informația inline despre stil.
- `title="..."` Furnizează mai multe informații despre un anumit element, spre deosebire de elementul `TITLE`, care creează un titlu pentru întreaga pagină Web.

La majoritatea elementelor se folosesc două atribute pentru a le internaționaliza.

- `lang="..."` Este identificatorul ISO de limbă.
- `dir="..."` Defineste directia textului (`ltr`, `rtl`)

Următoarele evenimente intrinseci se aplică la majoritatea elementelor:

- `onclick="..."` Arată că pe un dispozitiv de indicare (cum ar fi un mouse) s-a executat clic o dată.
- `ondblclick="..."` Arată că pe un dispozitiv de indicare (cum ar fi un mouse) s-a executat clic de două ori.
- `onmousedown="..."` Arată că pe un buton al mouse-ului s-a executat clic o dată și a fost ținut apăsat.
- `onmouseup="..."` Arată că un buton al mouse-ului, pe care s-a executat clic și a fost ținut apăsat, a fost eliberat.
- `onmouseover= "..."` Arată că un cursor a fost mutat cu ajutorul mouse-ului peste un obiect.
- `onmousemove="..."` Arată că mouse-ul a fost mutat.
- `onmouseout="..."` Arată că mouse-ul a deplasat cursorul în afara unui obiect.
- `onkeypress="..."` Arată că o tastă a fost apăsată și apoi eliberată.
- `onkeydown="..."` Arată că o tastă a fost apăsată și ținută apăsată.
- `onkeyup="..."` Arată că o tastă care a fost apăsată, a fost eliberată.

Anexa B
Tabelul de referințe pentru compararea browserelor

de Lee Anne Phillips

În această anexă

- **De ce este important un tabel de comparare a browserelor 734**
- **HTML azi 735**
- **Diferențele între browsere 735**
- **Tabelul de comparare a browserelor 736**
- **Evenimente intrinseci 799**
- **Entități HTML 4.0 de tip Escape 801**
- **Culori stabilite pentru HTML 4.0 801**
- **Entități HTML 4.0 de tip caracter 802**
- **Observații referitor la tabele 815**

World Wide Web reprezintă comunicare; dincolo de cacofonia năucitoare a diferitelor standarde, capacități de browser, limbi și stiluri de proiectare, încercăm să creăm canale de informare și divertisment între oameni și mașini, între producători și clienții lor, între furnizorii de informații și utilizatorii lor, între membrii unei comunități răspândiți în diverse locuri și între mașini și procese pe Web și utilizatorii lor umani. Această anexă încearcă să faciliteze acest lucru prin indicarea etichetelor și atributelor care sunt în mare măsură acceptate, precum și a celor care au o mai mică utilitate.

De ce este important un tabel de comparare a browserelor

De ce trebuie să vă pese de standardele HTML și care etichetă funcționează unde? Noul standard XML (Extensible Markup Language) nu va face el ca toate aceste lucruri să devină irelevante și depășite ?

Într-un singur cuvânt, nu. XML va mai întârzia și va dura chiar mai mult până va deveni un standard pe Web. Există miloane de pagini Web incorecte din punct de vedere sintactic, de fapt majoritatea paginilor sunt incorecte. XML depinde de capacitatea sa de a analiza un DTD pentru a înțelege cum este structurat un document, o idee bizară pentru oricine care a petrecut mult timp studiind codurile paginilor altor proiectanți, comparându-le cu perfecțiunea pură pe care a atins-o el însuși.

Corectarea tuturor paginilor HTML existente este un proiect care face ca problema anului 2000 să pară un fleac. Fiecare producător s-a lovit de simplul fapt că nu puteți forța un cod bun pe Web, trebuie să folosiți ceea ce există.

Dacă noul și strălucitorul dumneavoastră browser face ca o pagină să arate respingător după ce a funcționat bine ani întregi, ghiciți a cui este vina? Așa că ei au renunțat chiar la

analizarea limbajului HTML, optând pentru scoaterea etichetelor din „supa etichetelor" și încercând să facă ceva coerent cu ele. E mai greu decât pare.

HTML azi

Ce are un proiectant Web de făcut? Proliferarea etichetelor, limbajelor de scriptare inline, modulelor plug-in și a altor instrumente pentru crearea paginilor Web încep să facă din operația de scriere a codurilor pentru pagini mai mult o chestiune de programare decât o chestiune de proiectare, iar numărul divesificat de etichete devine o operație descurajantă de memorare, cu atât mai puțin de utilizare. Dimensiunea acestui tabel este un exemplu potrivit și nu include etichetele cu adevărat speciale WebTV sau alte extensii foarte specifice. Pe lângă intraneturile de firmă, unde puteți să fiți destul de siguri pe configurația de desktop, numărul de etichete și atribute acceptate de toate (sau aproape toate) platformele este încă destul de flexibil. Noile tehnologii se autoimpun în situri în loc să se suprapună, lăsând webmasterul să adauge el ceva deosebit sitului fără a crea probleme celorlalți.

În ciuda a ceea ce vă vor spune oamenii bine intenționați, noile standarde se vor modifica, chiar dacă numai puțin, în timpul unui proces de trecere de la o propunere de proiect la o recomandare și apoi la stadiul de RFC (Request for Comments), așa că nu trebuie să vă grăbiți. Nu contează care va fi auditoriul căruia intenționați să vă adresați, existența a milioane de PC-uri pe care sunt instalate programe și browsere vechi, cât și persoanele cu handicapuri sau alte segmente de piață importante, înseamă că pagina dumneavoastră trebuie să fie vizibilă și folositoare pentru o gamă mai largă de browsere decât cea existentă în biroul dumneavoastră. Cea mai sigură modalitate de verificare este de a vă asigura că paginile create pot fi citite într-un browser doar în mod text, cum ar fi browserul Lynx, chiar dacă pierde mult prin conversie. Având această ultimă soluție asigurată, puteți fi siguri că aproape toate browserele vor face ceva rezonabil cu pagina dumneavoastră.

Veți avea timpul să stăpâniți noile tehnologii, poate reinventând operația de proiectare pentru Web și creând spațiul atât pentru programatori cât și pentru artiștii grafici tradiționali și formând echipe pentru a combina ce este mai bun din ambele domenii.

Diferențele între browsere

Din nefericire, disputele între browsere s-au accentuat în ultima vreme. Microsoft și Netscape se luptă folosind diverse modele de obiecte, tratarea evenimentelor, etichete Dynamic HTML și alte incompatibilități, pentru a taxa ingeniozitatea și răbdarea webmasterului panicat. În principiu, aceste diferențe fac posibile comportările diferite ale celor două browsere.

În MSIE 4.0, de exemplu, evenimentele „se agită" prin obiectul asociat cel mai aproape cu un eveniment, spre fereastra părinte, iar conectarea datelor este dinamică. Aceasta înseamnă că fiecare element din pagină poate accesa evenimente cum ar fi `onClick`, oferind posibilitatea de a face ca pe aproape orice să poată fi executat clic sau să poată fi activat la trecerea cursorului peste el, inclusiv anteturile, dreptele orizontale și alte elemente care în mod obișnuit sunt statice. Mai înseamnă de asemenea că toată pagina poate fi din nou redată dinamic în orice moment folosind noile date. Ideea de bază a unei pagini este împinsă până la limită deoarece ea poate fi rescrisă cu noi informații, din mers, iar structura de bază este amorfă, ca un fel de robot din metal lichid din *Terminator 2*.

În Netscape Communicator și Navigator 4.0, în schimb, evenimentele sunt capturate mai întâi de fereastră. Numărul de elemente potențial interactive este restricționat mai mult sau mai puțin la imagini sau legături, acele elemente care au fost mult timp asociate, în mintea oamenilor, cu posibilitatea de realizare a unor acțiuni. După ce pagina a fost redată, ea este în principal statică, arătând mai mult ca o carte tipărită sau alte obiecte din viața de zi cu zi.

Ambele strategii au avantaje și dezavantaje, deși se pare că „agitarea" Microsoft, modelul de tip toate-elementele-navigabile, va învinge, potrivit ultimei idei de la consorțiul W3C, Document Object Model (DOM).

O pagină MSIE/W3C Document Object Model este fie o experiență multimedia bogată în facilități, fie un fel de „Hopa-Mitică" cu caractere care își modifică forma, „metamorfozându-se" în forme noi, în locul textului liniștit și previzibil cu care eram obișnuiți, în funcție de punctul de vedere al fiecăruia. Crearea unei pagini necesită atât perfecțiunea și restricțiile noilor modalități de gândire despre date, cât și mecanisme de afișare a paginilor.

O pagină Nescape arată mai mult ca o carte obișnuită (poate o carte de tip pop-up) dar cu posibilitatea de a avea aventuri multimedia unice ascunse la întoarcerea fiecărei pagini. Metodele tradiționale de afișare a paginii au fost extinse cu câteva etichete și atribute noi mai ușor de folosit și înțeles. Diferența constă în faptul că Microsoft/W3C Document Object Model a abandonat mai mult sau mai puțin „marcatoarele" tradiționale în favoarea legăturilor în afara paginii și „declanșatoarele" în favoarea interacțiunii cu aproape orice, fie că este vizibil sau nu, și, probabil, în favoarea controlării dinamice a fiecărui aspect al activității de navigare. Libertate și inovație față de responsabilitate și stabilitate: o confruntare clasică între generații.

Dar incompatibilitatea acestor două abordări, înseamnă că, până ce Netscape lansează un browser cu această facilitate, majoritatea paginilor vor folosi subseturi restrânse de caracteristici, care vor funcționa pe ambele platforme, iar proiectanții vor trebui să fie mai atenți la crearea unor pagini care să se adapteze fără probleme de la o platformă la alta, să învețe două (sau mai multe) metodologii în curs de elaborare și să fragmenteze în continuare lumea browserelor. Vor fi cu siguranță creatori de pagini care vor folosi focusul și vor forța vizitatorii să treacă printr-o învălmășeală „comercială" de efecte speciale înainte de a reda controlul utilizatorului. Hmmm...

Trăim vremuri interesante.

Tabelul de comparare a browserelor

- Tabelul este aranjat alfabetic, iar tipurile de etichete de structură/bloc/text sunt identificate după etichetă, astfel Ⓢ, Ⓑ și Ⓣ
- Etichetele și atributele scrise cu aldine sunt incluse în HTML 4.0, iar etichetele scrise între paranteze drepte sunt opționale.
- Valorile unor atribute sunt trecute între paranteze sub atributul corespunzător lor.
- Sunt folosite următoarele prescurtări:

  MSIE – Microsoft Internet Explorer
  Netscp – Netscape Navigator sau Communicator
  NCSA – NCSA Mosaic
  Opera – Opera
  Lynx – Lynx
  SM – Sun Microsystems
  Emacs – Emacs (modul W3C)
  HTML – Numărul licenței sau versiunii HTML
  Sponsor – Organizația de promulgare
  Status – Starea elementului în procesul de standardizare
  W3C – Consorțiul World Wide Web
  MS – Corporația Microsoft
  N – Corporația Netscape

Etichela	*MSIE*	*Netscp*	*NCSA*	*Opera*	*Lynx*	*Emacs*	*HTML*	*Sponsor*	*Status*
<!-- comment -- >	✔	✔	✔	✔	✔	✔	2	W3C	
ATENȚIE: Sintaxa comentariului SGML nu este un container de tip HTML, dar poate conține comentarii multiple încadrate între două liniuțe, un comentariu sau nimic. Primul comentariu, dacă există, trebuie să urmeze după semnul de exclamație și fără spații. Multe browsere nu implemetează acest element corect, așa că vă recomandăm să-l folosiți cu grijă. Evitați includerea etichetelor HTML și reduceți la minimum comentariile pe mai multe linii. Următoarele comentarii pot fi folosite:									
`<!>, <! -- x -- >, <! -- x --` `-- y -->`									
<! DOCTYPE>	✔	✔	✔	✔	✔	✔	W3C		
html	✔	✔	✔	✔	✔	✔	2	W3C	
<A></A> Ⓣ	✔	✔	✔	✔	✔	✔	2	W3C	
accesskey=	4	-	-	-	-	-	4	W3C	
charset=	4	-	-	-	-	-	4	W3C	
class=	-	✔3/4	-	-	✔*	✔	3/4	W3C	
coords=	4	-	-	-	-	-	4	W3C	
dir=	-	-	-	-	-	-	4	W3C	
href=	✔	✔	✔	✔	✔	✔	2-4	W3C	
ATENȚIE: Netscape a introdus un postfix `?subject=xxx` spre o adresă URL `mailto`, pentru a insera automat o linie de subiect în acele browsere care o acceptă, cum ar fi Netscape Navigator și MSIE, printre altele. Trebuie să aveți grijă, că această extensie este specifică doar câtorva browsere importante. Folosind acest tip de adresă URL și extensie poate determina să fie folosită o adresă poștală incorectă în alte browsere, cum ar fi NCSA Mozaic și Lynx, care sunt mai puțin folosite. Ar putea cauza chiar defectarea browserului.									

Etichetele și atributele care sunt scrise cu litere **aldine** sunt incluse în limbajul HTML 4.0, iar etichetele între paranteze drepte sunt opționale. Tabelul este aranjat în ordine alfabetică, pentru o regăsire mai ușoară, dar tipurile Structure/Block/Text ale etichetelor sunt identificate prin Ⓢ, Ⓑ, Ⓣ după eticheta propriu-zisă.

2,3,4 etc. = Numărul versiunii HTML condensate; 3+ = versiunea HTML 3.2; ✔ = disponibil pentru toate versiunile curente; ↓ nerecomandat – versiunea HTML; ✘ = depășit; Ⓟ = propriu/neconform standardului; *= implementare parțială sau parțial neconform standardului; NS=Non Specified.

Eticheta	*MSIE*	*Netscp*	*NCSA*	*Opera*	*Lynx*	*Emacs*	*HTML*	*Sponsor*	*Status*
<A></A> *continuare*									

Dacă vă preocupă accesibilitatea pe scară largă, puteți evita această caracteristică sau puteți construi linia subiectului cu atenție, astfel ca ea să funcționeze pe mai multe browsere. Pentru anumite aplicații, este foarte bine să cunoașteți automat de pe care pagină a trimis utilizatorul mesajul. Ca un exemplu, iată o legătură, care va funcționa pe majoritatea browserelor, deși linia de subiect poate fi sau nu completată:

```
E-mail Address: <A HREF="mailto:myname@mycompany.com?subject=Re:
➥myname@mycompany.com (LocationID)">myname@mycompany.com </A>
```

Orice cale ați alege, veți primi probabil indicații referitoare la locul de unde a pornit mesajul, fie din linia de subiect, fie ca un comentariu în adresa de e-mail, dacă este acceptată o legatură e-mail automată. Trebuie să rețineți formatul exact prezentat, deoarece este sensibil la schimbări, iar codul este inerent nesigur, fiind dependent de comportarea browserului, care se poate modifica în timp.

Acest format de legatură e-mail a fost testat în cadrul ultimelor versiuni de Netscape Navigator, MSIE, NCSA Mosaic, Opera și Lynx. Inserarea unei întreruperi de linie între ghilimele poate introduce o adresă deformată, dar totuși legală. Browserele care nu acceptă legături de tip `mailto` au un echivalent în text simplu, pe care îl pot copia în propriul lor program de e-mail.

id=	✔3/4	✔3/4	-	-	-	✔*	3/4	W3C	
lang=	✔3/4	✔3/4	-	-	-	✔*	3/4	W3C	
language=	✔3/4	✔3/4	-	-	-	-	Ⓟ	MS	✘3+
(javascript, vbscript)									
md=	-	-	-	-	✔*	✔	3	W3C	✘3+
methods=	✔2-4	✔	-	-	✔	✔	2	W3C	✘3+
name=	✔	✔	✔	✔	✔	✔	2-4	W3C	
rel=	✔	✔2-4	✔	✔	✔	✔	2-4	W3C	
rev=	-	✔2-4	✔	✔	✔	✔	2-4	W3C	
shape=	-	-	-	-	✔*	✔*	3/4	W3C	
style=	✔3/4	4	-	-	-	-	4	W3C	
tabindex=	✔3/4	4*	-	-	-	-	4	W3C	
target=	✔3/4	✔2-4	-	3	✔*	✔	4	W3C	
title=	✔2-4	✔2-4	✔	✔	✔	✔	2-4	W3C	
urn=	✔2-4	✔2-4	-	-	✔	✔	2	W3C	✘3+
event=	✔3/4	✔	-	✔	-	-	4	W3C	

Eticheta	*MSIE*	*Netscp*	*NCSA*	*Opera*	*Lynx*	*Emacs*	*HTML*	*Sponsor*	*Status*
<ABBREV></ABBREV> Ⓣ	-	-	-	-	✔*	✔	3	W3C	✘3+
class=	-	-	-	-	✔*	✔	3	W3C	✘3+
id=	-	-	-	-	✔	✔	3	W3C	✘3+
lang=	-	-	-	-	✔	✔	3	W3C	✘3+
<ACRONYM>... </ACRONYM> Ⓣ	-	-	-	-	✔*	✔	3/4	W3C	
class=	-	-	-	-	✔*	✔	3/4	W3C	
dir=	-	-	-	-	-	-	4	W3C	
id=	-	-	-	-	✔*	✔	3/4	W3C	
lang=	-	-	-	-	✔*	✔	3/4	W3C	
style=	-	-	-	-	✔*	✔	3/4	W3C	
title=	-	-	-	-	✔*	✔	3/4	W3C	
event=	-	-	-	-	-	-	3/4	W3C	
<ADDRESS>...</ADDRESS> Ⓣ	✔	✔	✔	✔	✔*	✔	2-4	W3C	
align=	✔3/4	-	-	-	-	-	Ⓟ		
(center, left, right)									
class=	-	3/4	-	-	✔*	✔	3/4	W3C	
clear=	-	✔3/4	-	-	✔*	✔	3	W3C	✘3+
(left, right, all)									
dir=	-	-	-	-	-	-	4	W3C	

Etichetele și atributele care sunt scrise cu litere **aldine** sunt incluse în limbajul HTML 4.0, iar etichetele între paranteze drepte sunt opționale. Tabelul este aranjat în ordine alfabetică, pentru o regăsire mai ușoară, dar tipurile Structure/Block/Text ale etichetelor sunt identificate prin Ⓢ, Ⓑ, Ⓣ după eticheta propriu-zisă.

2,3,4 etc. = Numărul versiunii HTML condensate; 3+ = versiunea HTML 3.2; ✔ = disponibil pentru toate versiunile curente; ↓ nerecomandat – versiunea HTML; ✘ = depășit; Ⓟ = propriu/neconform standardului; *= implementare parțială sau parțial neconform standardului; NS=Non Specified.

Eticheta	*MSIE*	*Netscp*	*NCSA*	*Opera*	*Lynx*	*Emacs*	*HTML*	*Sponsor*	*Status*
<ADDRESS></ADDRESS> *continuare*									
id=	✔	✔3/4	-	-	✔*	✔	3/4	W3C	
lang=	-	✔3/4	-	-	✔*	✔	3/4	W3C	
nowrap=	-	✔	-	-	✔*	✔	3	W3C	✘3+
style=	✔3/4	4	-	-	-	-	4	W3C	
title=	✔3/4	-	-	-	-	-	4	W3C	
event=	✔3/4	-	-	-	-	-	4	W3C	
<APP></APP>Ⓑ	4	-	-	-	-	-	Ⓟ	SM	
ATENȚIE: Eticheta <APP> este folosită de Sun Microsystems pentru HotJava, dar în general nu este acceptată în altă parte.									
class=	4	-	-	-	-	-	Ⓟ	SM	
src=	4	-	-	-	-	-	Ⓟ	SM	
align= (bottom, top, middle)	4	-	-	-	-	-	Ⓟ	SM	
width=	4	-	-	-	-	-	Ⓟ	SM	
height=	4	-	-	-	-	-	Ⓟ	SM	
<APPLET></APPLET>Ⓑ	✔3/4	✔2-4	-	3	✔*	✔	3+/4	W3C	↓4
align= (left, center, right, top, middle, bottom)	✔3/4	✔2-4	-	3	-	-	3+	W3C	
alt=	✔3/4	✔2-4	-	3	✔*	✔	3+	W3C	
archive=	-	-	-	-	-	-	4	W3C	
code=	✔3/4	✔2-4	-	✔	✔*	✔	3+	W3C	
codebase=	✔3/4	✔2-4	-	✔	✔*	✔	3+	W3C	
download=	✔3/4	-	-	-	-	-	Ⓟ	MS	✘3+
height=	✔3/4	✔2-4	-	3	✔*	✔	3+	W3C	
hspace=	✔3/4	✔2-4	-	3	✔*	✔	3+	W3C	

Eticheta	*MSIE*	*Netscp*	*NCSA*	*Opera*	*Lynx*	*Emacs*	*HTML*	*Sponsor*	*Status*
name=	✔3/4	✔2-4	-	3	✔*	✔	3+	W3C	
object=	-	4	-	-	-	-	4	W3C	
title=	✔3/4	-	-	-	-	-	3	MS	✘3+
vspace=	✔3/4	✔2-4	-	3	✔*	✔	3+	W3C	
width=	✔3/4	✔2-4	-	3	✔*	✔	3+	W3C	
<AREA>[</AREA>]Ⓑ	✔ Ⓟ	✔	✔	✔	✔*	✔	3+/4	W3C	
SUGESTIE: Browserul Lynx nu afișează imaginea, el creează o listă de referințe a tuturor destinațiilor într-o hartă imagine pe partea clientului. Dacă numele destinațiilor au sens, utilizatorii fără acces la hartă pot totuși naviga folosind modul text analog construit.									
alt=	-	✔2-4	✔	✔	✔*	✔	3+/4	W3C	
coords=	✔2-4	✔2-4	✔	✔	✔*	✔	3+/4	W3C	
href=	✔2-4	✔2-4	✔	✔	✔*	✔	3+/4	W3C	
id=	✔3/4	✔3	-	-	✔*	✔	3	W3C	✘3+
name=	✔3/4	-	-	-	✔*	✔	3	W3C	✘3+
nohref=	✔2-4	✔2-4	3	✔	✔*	✔	3+/4	W3C	
shape= (circ, circle, poly, polygon, rect, rectangle)	✔2-4	✔2-4	3	✔	✔*	✔	3+/4	W3C	
style=	✔3/4	4	-	-	-	-	3	W3C	✘3+
tabindex=	4	4*	-	-	-	-	4	W3C	
target=	✔3/4	✔2-4	-	-	-	-	4	W3C	

Etichetele și atributele care sunt scrise cu litere **aldine** sunt incluse în limbajul HTML 4.0, iar etichetele între paranteze drepte sunt opționale. Tabelul este aranjat în ordine alfabetică, pentru o regăsire mai ușoară, dar tipurile Structure/Block/Text ale etichetelor sunt identificate prin Ⓢ, Ⓑ, Ⓣ după eticheta propriu-zisă.

2,3,4 etc. = Numărul versiunii HTML condensate; 3+ = versiunea HTML 3.2; ✔ = disponibil pentru toate versiunile curente; ↓ nerecomandat – versiunea HTML; ✘ = depășit; Ⓟ = propriu/neconform standardului; *= implementare parțială sau parțial neconform standardului; NS=Non Specified.

Eticheta	*MSIE*	*Netscp*	*NCSA*	*Opera*	*Lynx*	*Emacs*	*HTML*	*Sponsor*	*Status*
<AREA>[</AREA>] *continuare*									
title=	✔3/4	✔2-4	-	-	-	-	3	W3C	✘3+
event=	✔3/4	-	-	-	-	-	3	W3C	✘3+
<AU></AU> T	-	-	-	-	✔*	✔	3	W3C	✘3+
class=	-	-	-	-	✔*	✔	3	W3C	
id=	-	-	-	-	✔*	✔	3	W3C	
lang=	-	-	-	-	✔*	✔	3	W3C	
<B></B> T	✔	✔	✔	✔	✔*	✔	2-4	W3C	
class=	-	✔3/4	-	-	✔*	✔	3/4	W3C	
dir=	-	-	-	-	-	-	4	W3C	
id=	✔3/4	✔3/4	-	-	✔*	✔	3/4	W3C	
lang=	-	✔3/4	-	-	✔*	✔	3/4	W3C	
style=	✔3/4	4	-	-	-	✔	4	W3C	
title=	✔3/4	4	-	-	-	-	4	W3C	
event=	✔3/4	-	-	-	-	-	4	W3C	
<BANNER></BANNER> B	-	-	-	-	✔*	✔	3	W3C	✘3+
class=	-	-	-	-	✔*	✔	3	W3C	
id=	-	-	-	-	✔*	✔	3	W3C	
lang=	-	-	-	-	✔*	✔	3	W3C	
<BASE>[</BASE>] S	✔	✔2-4	✔	✔	✔	✔	2-4	W3C	
	P	-	-	-	-	-			
href=	✔	✔2-4	✔	✔	✔	✔	2-4	W3C	
target=	✔3/4	✔2-4	-	3	✔*	✔	4	W3C	
title=	✔3/4	-	-	-	-	-	P	MS	

Eticheta	*MSIE*	*Netscp*	*NCSA*	*Opera*	*Lynx*	*Emacs*	*HTML*	*Sponsor*	*Status*
<BASEFONT> B ...[</BASEFONT>]	✔	✔	-	✔	✔*	✔	3+/4	W3C	↓4
color=	✔3/4	✔2-4	-	3	✔*	✔	3+/4	W3C	↓4
face=	✔3/4	✔2-4	-	3	✔*	✔	4	W3C	↓4
id=	✔3/4	✔3/4	-	-	✔*	✔	3	W3C	✘3+
size=	✔	✔2-4	-	✔	✔*	✔	3+	W3C	↓4
title=	✔3/4	-	-	-	-	-	-	MS	✘3+
<BDO> I	-	-	-	-	-	-	4	W3C	
dir=	-	-	-	-	-	-	4	W3C	
lang=	-	-	-	-	-	-	4	W3C	
<BGSOUND> B ...[</BGSOUND>]	✔	✔2-4	-	-	-	-	P	MS	✘3+
loop=	✔	✔2-4	-	-	-	-			
src=	✔	✔2-4	-	-	-	-			
id=	✔	-	-	-	-	-			
title=	✔	-	-	-	-	-			
<BIG></BIG> T	✔	✔	-	✔	✔*	✔	3+/4	W3C	
class=	-	✔3/4	-	-	✔*	✔	3/4	W3C	
dir=	-	-	-	-	-	-	3/4	W3C	
id=	✔3/4	✔3/4	-	-	✔*	✔	3/4	W3C	

Etichetele și atributele care sunt scrise cu litere **aldine** sunt incluse în limbajul HTML 4.0, iar etichetele între paranteze drepte sunt opționale. Tabelul este aranjat în ordine alfabetică, pentru o regăsire mai ușoară, dar tipurile Structure/Block/Text ale etichetelor sunt identificate prin S, B, T după eticheta propriu-zisă.

2,3,4 etc. = Numărul versiunii HTML condensate; 3+ = versiunea HTML 3.2; ✔ = disponibil pentru toate versiunile curente; ↓ nerecomandat – versiunea HTML; ✘ = depășit; P = propriu/neconform standardului; *= implementare parțială sau parțial neconform standardului; NS=Non Specified.

Eticheta	*MSIE*	*Netscp*	*NCSA*	*Opera*	*Lynx*	*Emacs*	*HTML*	*Sponsor*	*Status*
<BIG></BIG> *continuare*									
lang=	-	✔3/4	-	-	✔*	✔	3/4	W3C	
style=	✔3/4	4	-	-	-	-	4	W3C	
title=	✔3/4	-	-	-	-	-	4	W3C	
event=	✔3/4	-	-	-	-	-	4	W3C	
<BLINK></BLINK> (T)	-	✔2-4	-	-	-	(P)	NS	(P)	
<BLOCKQUOTE> (S) ... </BLOCKQUOTE>	✔	✔	✔	✔	✔	✔	2	W3C	
SUGESTIE: MSIE și Netscape tratează diferit „spațiul alb" din jurul unei secțiuni de text, în special când se imbrică blocurile. Nu se poate conta pe faptul că această etichetă va indenta sau deplasa textul de la marginea unui fundal, cu toate că este folosită adesea în acest scop.									
cite=	-	-	-	-	-	-	4	W3C	
class=	-	-	-	-	-	-	4	W3C	
dir=	-	-	-	-	-	-	4	W3C	
id=	3/4	✔3/4	-	-	-	-	4	W3C	
lang=	-	-	-	-	-	-	4	W3C	
style=	✔3/4	4	-	-	-	-	4	W3C	
title=	✔3/4	-	-	-	-	-	4	W3C	
event=	✔3/4	-	-	-	-	-	4	W3C	
[<BODY>][</BODY>] (S)	✔	✔	✔	✔	✔	✔	2-4	W3C	
OBSERVAȚIE: Eticheta <BODY> este opțională și poate fi dedusă.									
align= (center, left, right)	✔3/4	-	-	-	-	-	4	W3C	↓4
alink=	✔	✔	-	-	-	-	3+/4	W3C	↓4
background=	✔	✔	-	-	-	-	3+/4	W3C	↓4
bgcolor=	✔	✔	-	-	-	-	3+/4	W3C	↓4

Eticheta	*MSIE*	*Netscp*	*NCSA*	*Opera*	*Lynx*	*Emacs*	*HTML*	*Sponsor*	*Status*
bgproperties= (fixed)	✔3/4	-	-	-	-	-	(P)	MS	✘3+
class=	-	-	-	-	-	-	4	W3C	
dir=	-	-	-	-	-	-	4	W3C	
id=	✔3/4	✔3/4	-	-	-	-	4	W3C	
lang=	-	-	-	-	-	-	4	W3C	
leftmargin=	✔	-	-	-	-	-	(P)	MS	
link=	✔3/4	✔2-4	-	-	-	-	3+/4	W3C	↓4
scroll=	✔3/4	-	-	-	-	-	(P)	MS	
style=	✔3/4	4	-	-	-	-	4	W3C	
text=	✔3/4	✔2-4	-	-	-	-	3+/4	W3C	↓4
title=	✔3/4	-	-	-	-	-	4	W3C	
topmargin=	✔3/4	-	-	-	-	-	(P)	MS	
vlink=	✔3/4	✔2-4	-	-	-	-	3+/4	W3C	↓4
event=	✔3/4	-	-	-	-	-	4	W3C	
  (S)(T)	✔	✔	✔	✔	✔	✔	2-4	W3C	
ATENȚIE: Metoda inserării mai multor etichete sau <P> pentru a insera „spațiu alb" este o metodă nestandard și poate conduce la erori. Standardul HTML poate fi interpretat în sensul ca excesul de spațiu alb să fie ignorat. Unele browsere ignoră etichetele multiple, deși majoritatea producătorilor de browsere au interpretat standardul astfel ca fiecare etichetă să insereze o altă linie.									
class=	✔	✔3/4	-	-	-	-	3	W3C	

Etichetele și atributele care sunt scrise cu litere aldine sunt incluse în limbajul HTML 4.0, iar etichetele între paranteze drepte sunt opționale. Tabelul este aranjat în ordine alfabetică, pentru o regăsire mai ușoară, dar tipurile Structure/Block/Text ale etichetelor sunt identificate prin (S), (B), (T) după eticheta propriu-zisă.

2,3,4 etc. = Numărul versiunii HTML condensate; 3+ = versiunea HTML 3.2; ✔ = disponibil pentru toate versiunile curente; ↓ nerecomandat – versiunea HTML; ✘ = depășit; (P) = propriu/neconform standardului; *= implementare parțială sau parțial neconform standardului; NS=Non Specified.

Eticheta	*MSIE*	*Netscp*	*NCSA*	*Opera*	*Lynx*	*Emacs*	*HTML*	*Sponsor*	*Status*
  continuare									
clear=	✔	✔2-4	-	-	-	-	3	W3C	
(left, right, all)									
none							3+	W3C	
dir=	-	-	-	-	-	-	4	W3C	
id=	✔	✔3/4	-	-	-	-	3	W3C	
lang=	-	✔3/4	-	-	-	-	3	W3C	
style=	✔3/4	4	-	-	-	-	4	W3C	
title=	✔3/4	-	-	-	-	-	4	W3C	
<BQ></BQ> [S]	-	-	-	-	✔	✔	3	W3C	✘3+
ATENȚIE: <BQ> este denumirea sinonimă din limbajul HTML 3.0 pentru <BLOCKQUOTE> dar nu prea mai este acceptată.									
<BUTTON></BUTTON> [S]	✔3/4	-	-	-	-	-	4	W3C	
accesskey=	✔3/4	-	-	-	-	-	[P]	MS	
class=	4	-	-	-	-	-	4	W3C	
dir=	-	-	-	-	-	-	4	W3C	
disabled=	✔3/4	-	-	-	-	-	4	W3C	
id=	4	-	-	-	-	-	4	W3C	
lang=	-	-	-	-	-	-	4	W3C	
name=	-	-	-	-	-	-	4	W3C	
style=	4	-	-	-	-	-	4	W3C	
tabindex=	4	-	-	-	-	-	4	W3C	
title=	✔3/4	-	-	-	-	-	4	W3C	
type=	4	-	-	-	-	-	4	W3C	
value=	4	-	-	-	-	-	4	W3C	
event=	✔3/4	-	-	-	-	-	4	W3C	

Eticheta	*MSIE*	*Netscp*	*NCSA*	*Opera*	*Lynx*	*Emacs*	*HTML*	*Sponsor*	*Status*
<CAPTION>... </CAPTION> [S]	✔	✔	✔	✔	✔	✔	3	W3C	
align=	✔	✔	-	✔	✔	✔	3+/4	W3C	
(center, left, right)									
(top, bottom)	✔	-	-	-	-	-	4	W3C	
class=	-	-	-	-	-	-	4	W3C	
dir=	-	-	-	-	-	-	4	W3C	
id=	✔	✔3/4	-	-	✔	✔	3/4	W3C	
lang=	-	-	-	-	✔	✔	3/4	W3C	
style=	✔3/4	4	-	-	-	-	4	W3C	
title=	✔3/4		-	-	-	-	4	W3C	
valign=	✔2-4	✔	✔	-	✔	✔	3+	W3C	✘4
(top, bottom)									
event=	✔3/4	-	-	-	-	-	4	W3C	
<CENTER></CENTER> [S]	✔	✔	✔	✔	✔	✔	3+	W3C	↓4
OBSERVAȚIE: Standardul HTML 3.2 definea eticheta <CENTER> ca o scriere scurtată pentru <DIV ALIGN="CENTER"> deși folosirea ei este nerecomandată acum.									
class=	-	-	-	-	-	-	4	W3C	
dir=	-	-	-	-	-	-	4	W3C	
id=	✔3/4	✔3/4	-	-	✔	✔	3	W3C	
lang=	-	-	-	-	-	-	4	W3C	

Etichetele și atributele care sunt scrise cu litere **aldine** sunt incluse în limbajul HTML 4.0, iar etichetele între paranteze drepte sunt opționale. Tabelul este aranjat în ordine alfabetică, pentru o regăsire mai ușoară, dar tipurile Structure/Block/Text ale etichetelor sunt identificate prin [S], [B], [T] după eticheta propriu-zisă.

2,3,4 etc. = Numărul versiunii HTML condensate; 3+ = versiunea HTML 3.2; ✔ = disponibil pentru toate versiunile curente; ↓ nerecomandat – versiunea HTML; ✘ = depășit; [P] = propriu/neconform standardului; *= implementare parțială sau parțial neconform standardului; NS=Non Specified.

Eticheta	*MSIE*	*Netscp*	*NCSA*	*Opera*	*Lynx*	*Emacs*	*HTML*	*Sponsor*	*Status*
<CENTER></CENTER> *continuare*									
style=	✔3/4	4	-	-	-	-	4	W3C	
title=	✔3/4	-	-	-	-	-	4	W3C	
event=	✔3/4	-	-	-	-	-	4	W3C	
<CITE></CITE> Ⓣ	✔	✔	✔	✔	✔*	✔	2-4	W3C	
class=	-	✔3/4	-	-	✔	✔	3/4	W3C	
dir=	-	-	-	-	-	-	4	W3C	
id=	✔3/4	✔3/4	-	-	✔	✔	3/4	W3C	
lang=	-	✔3/4	-	-	✔	✔	3/4	W3C	
style=	✔3/4	4	-	-	-	-	4	W3C	
title=	✔3/4	-	-	-	-	-	4	W3C	
event=	✔3/4	-	-	-	-	-	4	W3C	
<CODE></CODE> Ⓣ	✔	✔	✔	✔	✔*	✔	2	W3C	
class=	-	✔3/4	-	-	✔	✔	3	W3C	
dir=	-	-	-	-	-	-	4	W3C	
id=	✔3/4	✔3/4	-	-	✔	✔	3	W3C	
lang=	-	✔3/4	-	-	✔	✔	3	W3C	
style=	✔3/4	4	-	-	-	-	4	W3C	
title=	✔3/4	-	-	-	-	-	4	W3C	
event=	✔3/4	-	-	-	-	-	4	W3C	
<COL></COL> Ⓢ	4	-	-	-	-	-	4	W3C	
align= (center, left, right)	4	-	-	-	-	-	4	W3C	
char=	-	-	-	-	-	-	4	W3C	
charoff=	-	-	-	-	-	-	4	W3C	

Eticheta	*MSIE*	*Netscp*	*NCSA*	*Opera*	*Lynx*	*Emacs*	*HTML*	*Sponsor*	*Status*
class=	-	-	-	-	-	-	4	W3C	
dir=	-	-	-	-	-	-	4	W3C	
id=	4	-	-	-	-	-	4	W3C	
span=	4	-	-	-	-	-	4	W3C	
style=	4	-	-	-	-	-	4	W3C	
title=	4	-	-	-	-	-	4	W3C	
valign= (baseline, bottom, center, top)	4	-	-	-	-	-	4	W3C	
width=	4	-	-	-	-	-	4	W3C	
event=	4	-	-	-	-	-	4	W3C	
<COLGROUP> Ⓢ	✔3/4	-	-	-	-	-	4	W3C	
align= (center, left, right)	4	-	-	-	-	-	4	W3C	
char=	-	-	-	-	-	-	4	W3C	
charoff=	-	-	-	-	-	-	4	W3C	
class=	-	-	-	-	-	-	4	W3C	
id=	4	-	-	-	-	-	4	W3C	
span=	4	-	-	-	-	-	4	W3C	
style=	4	-	-	-	-	-	4	W3C	

Etichetele și atributele care sunt scrise cu litere **aldine** sunt incluse în limbajul HTML 4.0, iar etichetele între paranteze drepte sunt opționale. Tabelul este aranjat în ordine alfabetică, pentru o regăsire mai ușoară, dar tipurile Structure/Block/Text ale etichetelor sunt identificate prin Ⓢ, Ⓑ, Ⓣ după eticheta propriu-zisă.

2,3,4 etc. = Numărul versiunii HTML condensate; 3+ = versiunea HTML 3.2; ✔ = disponibil pentru toate versiunile curente; ↓ nerecomandat – versiunea HTML; ✘ = depășit; Ⓟ = propriu/neconform standardului; *= implementare parțială sau parțial neconform standardului; NS=Non Specified.

Eticheta	*MSIE*	*Netscp*	*NCSA*	*Opera*	*Lynx*	*Emacs*	*HTML*	*Sponsor*	*Status*
<COLGROUP> *continuare*									
title=	4	-	-	-	-	-	4	W3C	
valign= (baseline, bottom, center, top)	-	-	-	-	-	-	4	W3C	
width=	4	-	-	-	-	-	4	W3C	
<COMMENT> Ⓣ ... </COMMENT>	✔	-	-	-	✔	-	Ⓟ		✘2
title=	✔	-	-	-	-	-			
ATENȚIE : Orice informație inserată între etichetele <COMMENT> de început și de sfârșit este tratată ca un comentariu de către MSIE și Lynx, dar etichetele sunt ignorate de majoritatea celorlalte browsere, așa că textul sau codul plasat în interiorul lor este redat ca și cum etichetele nu ar fi acolo.									
<CREDIT></CREDIT> Ⓣ	-	-	-	-	✔*	✔	3	W3C	
<DD> Ⓢ	✔	✔	✔	✔	✔	✔	2-4	W3C	
align= (center, left, right)	4	-	-	-	-	-	3	W3C	✘4
class=	4	3	-	-	-	-	4	W3C	
dir=	-	-	-	-	-	-	4	W3C	
id=	4	3	-	-	-	-	3	W3C	✘4
lang=	-	3	-	-	-	-	3	W3C	✘4
style=	4	4	-	-	-	-	3	W3C	✘4
title=	4	-	-	-	-	-	3	W3C	✘4
event=	4	-	-	-	-	-	-	-	✘4
<DEL></DEL> Ⓟ	-	-	-	-	✔*	✔	3	W3C	✘3+
cite=	-	-	-	-	-	-	4	W3C	
class=	-	-	-	-	✔	✔	¾	W3C	
datetime=	-	-	-	-	-	-	4	W3C	

Eticheta	*MSIE*	*Netscp*	*NCSA*	*Opera*	*Lynx*	*Emacs*	*HTML*	*Sponsor*	*Status*
dir=	-	-	-	-	-	-	4	W3C	
id=	-	-	-	-	✔	✔	3/4	W3C	
lang=	-	-	-	-	✔	✔	3/4	W3C	
style=	-	-	-	-	✔	✔	3/4	W3C	
title=	-	-	-	-	✔	✔	3/4	W3C	
event=	-	-	-	-	-	-	3/4	W3C	
<DFN></DFN> Ⓣ	✔	✔	✔	✔	✔*	✔	3/4	W3C	
class=	-	-	-	-	✔	✔	3/4	W3C	
dir=	-	-	-	-	-	-	4	W3C	
id=	✔3/4	-	-	-	✔	✔	3/4	W3C	
lang=	-	-	-	-	✔	✔	3/4	W3C	
style=	4	-	-	-	-	-	4	W3C	
title=	4	-	-	-	-	-	4	W3C	
event=	4	-	-	-	-	-	4	W3C	
<DIR></DIR> Ⓢ	✔	✔	✔	✔	✔	✔	2	W3C	↓4
class=	-	-	-	-	-	-	4	W3C	
compact	✔	✔	✔	✔	✔*	✔	2	W3C	↓4
dir=	-	-	-	-	-	-	4	W3C	
id=	4	✔3/4	-	-	-	-	4	W3C	

Etichetele și atributele care sunt scrise cu litere **aldine** sunt incluse în limbajul HTML 4.0, iar etichetele între paranteze drepte sunt opționale. Tabelul este aranjat în ordine alfabetică, pentru o regăsire mai ușoară, dar tipurile Structure/Block/Text ale etichetelor sunt identificate prin Ⓢ, Ⓑ, Ⓣ după eticheta propriu-zisă.

2,3,4 etc. = Numărul versiunii HTML condensate; 3+ = versiunea HTML 3.2; ✔ = disponibil pentru toate versiunile curente; ↓ nerecomandat – versiunea HTML; ✘ = depășit; Ⓟ = propriu/neconform standardului; *= implementare parțială sau parțial neconform standardului; NS=Non Specified.

Eticheta	*MSIE*	*Netscp*	*NCSA*	*Opera*	*Lynx*	*Emacs*	*HTML*	*Sponsor*	*Status*
<DIR></DIR> *continuare*									
lang=	-	-	-	-	-	-	3/4	W3C	
style=	4	4	-	-	-	-	4	W3C	
title=	4	-	-	-	-	-	4	W3C	
event=	4	-	-	-	-	-	4	W3C	
<DIV> Ⓢ	✔	✔	✔	✔	✔	✔	3/4	W3C	
align=	4	✔	✔	✔	✔	✔	3/4	W3C	
(center, left, right)	✔	-	-	-	-	-	3/4		
(justify)	-	-	-	-	-	-	3	W3C	✘3+
class=	-	✔3/4	-	-	✔	✔	3/4	W3C	
clear=	-	✔	-	-	✔*	✔	3	W3C	✘3+
(left, right, all)	✔	-	-	✔*	✔	-			
datafld=	4	-	-	-	-	-	Ⓟ		
dataformats=	4	-	-	-	-	-	Ⓟ		
datasrc=	4	-	-	-	-	-	Ⓟ		
dir=	-	-	-	-	-	-	4	W3C	
id=	4	✔3/4	-	-	✔	✔	3/4	W3C	
lang=	-	✔3/4	-	-	✔	✔	3/4	W3C	
nowrap=	-	✔3/4	-	-	✔	✔	3	W3C	✘3+
style=	4	4	-	-	-	-	4	W3C	
title=	4	-	-	-	-	-	4	W3C	
event=	4	-	-	-	-	-	4	W3C	
<DL> Ⓢ	✔	✔	✔	✔	✔	✔	2-4	W3C	
align=	4	-	-	-	-	-			
(center, left, right)									

Eticheta	*MSIE*	*Netscp*	*NCSA*	*Opera*	*Lynx*	*Emacs*	*HTML*	*Sponsor*	*Status*
class=	-	3	-	-	✔	✔	3/4	W3C	
clear=	-	✔	-	-	✔*	✔	3	W3C	
(left, right, all)	✔	-	-	✔*	✔	-			
compact=	✔	✔	✔	✔	✔	✔	2-4	W3C	
dir=	-	-	-	-	-	-	4	W3C	
id=	4	✔3/4	-	-	✔	✔	3/4	W3C	
lang=	-	✔3/4	-	-	-	-	3/4	W3C	
style=	4	4	-	-	-	-	4	W3C	
title=	4	-	-	-	-	-	4	W3C	
event=	4	-	-	-	-	-	4	W3C	
<DT> Ⓢ	✔	✔	✔	✔	✔	✔	2-4	W3C	
align=	4	-	-	-	-	-	3	W3C	✘3+
(center, left, right)									
class=	-	-	-	-	-	-	4		
dir=	-	-	-	-	-	-	4		
id=	4	-	-	-	✔	✔	3	W3C	
lang=	-	-	-	-	-	-	4	W3C	
style=	4	-	-	-	-	-	4	W3C	
title=	4	-	-	-	-	-	4	W3C	
event=	4	-	-	-	-	-	4	W3C	

Etichetele și atributele care sunt scrise cu litere **aldine** sunt incluse în limbajul HTML 4.0, iar etichetele între paranteze drepte sunt opționale. Tabelul este aranjat în ordine alfabetică, pentru o regăsire mai ușoară, dar tipurile Structure/Block/Text ale etichetelor sunt identificate prin Ⓢ, Ⓑ, Ⓣ după eticheta propriu-zisă.

2,3,4 etc. = Numărul versiunii HTML condensate; 3+ = versiunea HTML 3.2; ✔ = disponibil pentru toate versiunile curente; ↓ nerecomandat – versiunea HTML; ✘ = depășit; Ⓟ = propriu/neconform standardului; *= implementare parțială sau parțial neconform standardului; NS=Non Specified.

Eticheta	MSIE	Netscp	NCSA	Opera	Lynx	Emacs	HTML	Sponsor	Status
<EM></EM> Ⓣ	✔	✔	✔	✔	✔*	✔	2-4	W3C	
class=	-	-	-	-	✔	✔	3/4	W3C	
dir=	-	-	-	-	-	-	4	W3C	
id=	✔3/4	-	-	-	-	-	3/4	W3C	
lang=	-	-	-	-	-	-	3/4	W3C	
style=	4	-	-	-	-	-	4	W3C	
title=	4	-	-	-	-	-	4	W3C	
event=	4	-	-	-	-	-	4	W3C	
<EMBED></EMBED> Ⓑ	✔3/4	✔	-	-	-	-	Ⓟ	NS	

OBSERVAȚIE: Eticheta <EMBED> a fost inclusă în MSIE pentru compatibilitatea cu versiuni anterioare. Folosiți eticheta <OBJECT> dacă este posibil.

accesskey=	4	-	-	-	-	-			
align=	4	✔	-	-	-	-			
(absbottom, absmiddle, baseline, bottom, left, middle, right, texttop, top)									
height=	4	-	-	-	-	-			
hidden=	4	-	-	-	-	-			
id=	4	-	-	-	-	-			
palette=	4	-	-	-	-	-			
pluginspage=	4	-	-	-	-	-			
src=	4	-	-	-	-	-			
style=	4	-	-	-	-	-			
title=	4	-	-	-	-	-			
width=	4	-	-	-	-	-			
event=	4	-	-	-	-	-			

Eticheta	MSIE	Netscp	NCSA	Opera	Lynx	Emacs	HTML	Sponsor	Status
<FIELDSET>...</FIELDSET> Ⓢ	4	-	-	-	-	-	4	W3C	
class=	-	-	-	-	-	-	4	W3C	
dir=	-	-	-	-	-	-	4	W3C	
id=	-	-	-	-	-	-	4	W3C	
lang=	-	-	-	-	-	-	4	W3C	
style=	-	-	-	-	-	-	4	W3C	
title=	-	-	-	-	-	-	4	W3C	
event=	-	-	-	-	✔*	✔	3	W3C	✘3+
<FIG></FIG> Ⓢ	-	-	-	-	✔*	✔	3	W3C	✘3+
align=									
(left, center, right)							3	W3C	✘3+
(justify, bleedleft, bleedright)					✔	✔	3	W3C	✘3+
class=	-	-	-	-	✔	✔	3	W3C	✘3+
clear=									
(left, right, all)							3	W3C	✘3+
height=	-	-	-	-	-	-	3	W3C	✘3+
id=	-	-	-	-	-	-	3	W3C	✘3+
imagemap=	-	-	-	-	-	-	3	W3C	✘3+
lang=	-	-	-	-	-	-			

Etichetele și atributele care sunt scrise cu litere **aldine** sunt incluse în limbajul HTML 4.0, iar etichetele între paranteze drepte sunt opționale. Tabelul este aranjat în ordine alfabetică, pentru o regăsire mai ușoară, dar tipurile Structure/Block/Text ale etichetelor sunt identificate prin Ⓢ, Ⓑ, Ⓣ după eticheta propriu-zisă.

2,3,4 etc. = Numărul versiunii HTML condensate; 3+ = versiunea HTML 3.2; ✔ = disponibil pentru toate versiunile curente; ↓ nerecomandat – versiunea HTML; ✘ = depășit; Ⓟ = propriu/neconform standardului; *= implementare parțială sau parțial neconform standardului; NS=Non

Eticheta	*MSIE*	*Netscp*	*NCSA*	*Opera*	*Lynx*	*Emacs*	*HTML*	*Sponsor*	*Status*
<FIG></FIG> *continuare*									
md=	-	-	-	-	-	-	3	W3C	✘3+
noflow	-	-	-	-	-	-	3	W3C	✘3+
src=	-	-	-	-	-	-	3	W3C	✘3+
units=	-	-	-	-	-	-	3	W3C	✘3+
width=	-	-	-	-	-	-	3	W3C	✘3+
<FN></FN> Ⓢ	-	-	-	-	✔	✔	3	W3C	✘3+
class=	-	-	-	-	✔	✔	3	W3C	
dir=	-	-	-	-	-	-	4	W3C	
id=	-	-	-	-	✔	✔	3	W3C	
lang=	-	-	-	-	✔	✔	3	W3C	
<FONT></FONT> Ⓣ	✔3/4	✔	-	✔	✔*	✔	3+	W3C	↓4
ATENȚIE: Folosirea etichetei `<FONT>` poate determina o comportare anormală a browserelor care folosesc un set de caractere ne-Roman sau care foosesc un set de fonturi diferite de cele care sunt configurate pe calculatorul proiectantului. Folosiți CSS dacă este posibil.									
color=	✔	✔	-	✔	✔*	✔	3+	W3C	↓4
face=	✔	-	-	-	-	-	4	W3C	↓4
id=	-	-	-	-	-	-	-		✘3+
size=	✔	✔	-	✔	✔*	✔	3+	W3C	↓4
style=	-	-	-	-	-	-	-	W3C	✘3+
title=	-	-	-	-	-	-	-	W3C	✘3+
event=	-	-	-	-	-	-	-	W3C	✘3+
<FORM></FORM> Ⓢ	✔	✔	✔	✔	✔*	✔	2-4	W3C	
accept-charset=	-	-	-	-	-	-	4	W3C	
action=	✔	✔	✔	✔	✔	✔	2-4	W3C	
class=	-	-	-	-	-	-	4	W3C	

Eticheta	*MSIE*	*Netscp*	*NCSA*	*Opera*	*Lynx*	*Emacs*	*HTML*	*Sponsor*	*Status*
dir=	-	-	-	-	-	-	4	W3C	
enctype=	✔	✔	✔	✔	✔	✔	2-4	W3C	
ATENȚIE: Unele browsere mai vechi nu acceptă atributul `enctype`, care transformă conținuturile într-un tip prestabilit, de obicei acesta este `enctype="application/x-www-form-urlencoded"` sau o variantă mai apropiată. Textul obținut trebuie să fie decodat de un script CGI sau alt script de pe server înainte de procesare. Unele browsere permit ca atributul `enctype="text/plain"` să trimită datele sub formă de text simplu. Această metodă poate fi folosită când se folosește caracteristica `mailto`: pentru a evita nevoia unui script CGI pe server; totuși dacă nu este tratat corect, poate cauza probleme de siguranță și corupere a datelor.									
În plus, unele browsere acceptă atributul `enctype="application/sgml-form-urlencoded"`, care este identic cu atributul perstabilit `enctype` menționat anterior, dar separatorii dintre perechile nume/valoare sunt punct și virgulă (;) în loc de ampersand (&). Acest format este preferat când se folosește `method="GET"` deoarece folosirea semnelor ampersand poate duce la coruperea datelor transferate în anumite împrejurări.									
Netscape a introdus tipuri suplimentare de date pentru a susține caracteristicile „îmbunătățite" care nu sunt acceptate de alte browsere. Când folosiți aceste tipuri de date, trebuie să fiți atenți să nu creați probleme altor browsere. În particular, folosirea unui atribut `enctype="multipart/form-data"` cu o etichetă `<INPUT>` care cere un nume de fișier poate crea probleme. Deși MSIE și altele adaugă acest suport, tratarea corectă a acestui tip de date nu este universală.									
id=	4	-	-	-	-	-	4	W3C	
lang=	-	-	-	-	-	-	4	W3C	
method=	✔	✔	✔	✔	✔	✔	2-4	W3C	
(get, post)	✔	✔	✔	✔	✔	✔	2-4	W3C	
name=	4	-	-	-	-	-	-		✘3+
script=	-	-	-	-	-	-	3	W3C	✘3+
style=	4	-	-	-	-	-	4	W3C	
target=	4	-	-	-	-	-	-	W3C	

Etichetele și atributele care sunt scrise cu litere **aldine** sunt incluse în limbajul HTML 4.0, iar etichetele între paranteze drepte sunt opționale. Tabelul este aranjat în ordine alfabetică, pentru o regăsire mai ușoară, dar tipurile Structure/Block/Text ale etichetelor sunt identificate prin Ⓢ, Ⓑ, Ⓣ după eticheta propriu-zisă.

2,3,4 etc. = Numărul versiunii HTML condensate; 3+ = versiunea HTML 3.2; ✔ = disponibil pentru toate versiunile curente; ↓ nerecomandat – versiunea HTML; ✘ = depășit; Ⓟ = propriu/neconform standardului; *= implementare parțială sau parțial neconform standardului; NS=Non Specified.

Eticheta	*MSIE*	*Netscp*	*NCSA*	*Opera*	*Lynx*	*Emacs*	*HTML*	*Sponsor*	*Status*
<FORM></FORM> *continuare*									
title=	4	-	-	-	-	-	4	W3C	
event=	4	-	-	-	-	-	4	W3C	
<FRAME></FRAME> Ⓢ	✔	✔	-	✔	✔*	✔	4	W3C	
SUGESTIE: Eticheta `<FRAME>` este încă controversată și ar trebui să fie folosită cu grijă, în special deoarece afectează accesibilitatea pentru utilizatorii care folosesc browsere în text Braille sau audio. Folosiți întotdeana containerul `<NOFRAMES>` pentru a include un conținut cu sens, cum ar fi o listă de elemente de navigare obișnuite, pentru a permite folosirea pe toate browserele. Inserarea unui mesaj concis care să le spună vizitatorilor să folosească un alt browser, mai „modern", poate fi o metodă cam prea dură.									
bordercolor=	✔	✔3/4	-	-	-	-	-		✘4
class=	-	-	-	-	-	-	4	W3C	
dir=	-	-	-	-	-	-	4	W3C	
frameborder=	✔3/4	4	-	-	-	-	4	W3C	
framespacing=	✔3/4	✔	-	-	-	-	-		✘4
height=	4	-	-	-	-	-	-	W3C	✘4
id=	4	-	-	-	-	-	4	W3C	
marginheight=	✔3/4	✔	-	-	-	-	4	W3C	
marginwidth=	✔3/4	✔	-	-	-	-	4	W3C	
method=	-	-	-	-	-	-	4	W3C	
name=	✔3/4	✔2-4	-	✔	✔*	✔	4	W3C	
(window_name, _blank, _parent, _self, _top)									
noresize=	✔3/4	✔	-	-	-	-	4	W3C	
(noresize, resize)									
scrolling=	✔3/4	✔2-4	-	-	-	-	Ⓟ	MS	✘4
(auto, no, yes)									
src=	✔3/4	✔2-4	-	✔	✔*	✔	4	W3C	
style	-	-	-	-	-	-	-	W3C	✘4
target=	✔	✔	-	✔	✔*	✔	4	W3C	

Eticheta	*MSIE*	*Netscp*	*NCSA*	*Opera*	*Lynx*	*Emacs*	*HTML*	*Sponsor*	*Status*
title=	4	-	-	-	-	-	4	W3C	
width=	4	-	-	-	-	-	-	W3C	✘4
event=	4	-	-	-	-	-	4	W3C	
<FRAMESET>...</FRAMESET> Ⓢ	✔3/4	✔2-4	-	✔	✔*	✔	4	W3C	
border=	4	-	-	-	-	-	-		✘4
bordercolor=	4	-	-	-	-	-	-		✘4
cols=	4	✔	-	-	-	-	4	W3C	
frameborder=	4	-	-	-	-	-	-		✘4
framespacing=	4	-	-	-	-	-	-		✘4
id=	4	-	-	-	-	-	-	W3C	✘4
rows=	4	✔	-	-	-	-	4	W3C	
title=	4	-	-	-	-	-	-	W3C	✘4
event=	4	-	-	-	-	-	4	W3C	
(onload, onunload)									
[<HEAD>][</HEAD>] Ⓢ	✔	✔	✔	✔	✔	✔	2-4	W3C	
OBSERVAȚIE : Eticheta `<HEAD>` este opțională și poate fi dedusă. Deși etichetele `<BASE>`, `<ISINDEX>`, `<LINK>` și `<META>` nu sunt containere în titlu, Microsoft și probabil și alții permit încheierea etichetelor container. Această practică este interzisă prin specificații, dar deoarece nu afectează comportarea în nici un fel, nu prea contează dacă folosiți încheierea etichetelor container.									
id=	4	-	-	-	-	-	-	W3C	✘4
dir=	-	-	-	-	-	-	4	W3C	

Etichetele și atributele care sunt scrise cu litere **aldine** sunt incluse în limbajul HTML 4.0, iar etichetele între paranteze drepte sunt opționale. Tabelul este aranjat în ordine alfabetică, pentru o regăsire mai ușoară, dar tipurile Structure/Block/Text ale etichetelor sunt identificate prin Ⓢ, Ⓑ, Ⓣ după eticheta propriu-zisă.

2,3,4 etc. = Numărul versiunii HTML condensate; 3+ = versiunea HTML 3.2; ✔ = disponibil pentru toate versiunile curente; ↓ nerecomandat – versiunea HTML; ✘ = depășit; Ⓟ = propriu/neconform standardului; *= implementare parțială sau parțial neconform standardului; NS=Non Specified.

Eticheta	*MSIE*	*Netscp*	*NCSA*	*Opera*	*Lynx*	*Emacs*	*HTML*	*Sponsor*	*Status*
[<HEAD>][</HEAD>] Ⓢ *continuare*									
lang=	-	-	-	-	-	-	4	W3C	
profile=	-	-	-	-	-	-	4	W3C	
style=	4	-	-	-	-	-	-	W3C	✗4
title=	4	-	-	-	-	-	4	W3C	
<H1></H1> Ⓢ	✔	✔	✔	✔	✔*	✔	2-4	W3C	
align=	✔	✔	✔	✔	✔	✔	3	W3C	↓4
(left, center, right)									
(justify)							3	W3C	✗3
class=	-	-	-	-	-	-	3/4	W3C	
clear=	✔	✔	-	-	-	-	3	W3C	
(left, right, all)									
color=	✔	-	-	-	-	-	Ⓟ		✗3+
dingbat=	-	-	-	-	✔*	✔	3	W3C	✗3+
dir=	-	-	-	-	-	-	4	W3C	
id=	-	-	-	-	✔	✔	3/4	W3C	
lang=	-	-	-	-	✔	✔	3/4	W3C	
md=	-	-	-	-	✔	✔	3	W3C	✗3*
nowrap	-	-	-	-	✔	✔	3	W3C	✗3+
seqnum=	-	-	-	-	-	-	3	W3C	✗3+
skip=	-	-	-	-	-	-	3	W3C	
style=	-	-	-	-	-	-	4	W3C	
title=	-	-	-	-	-	-	4	W3C	
<H2>, <H3>, <H4>, <H5>, <H6> Identic cu **<H1>**									

Eticheta	*MSIE*	*Netscp*	*NCSA*	*Opera*	*Lynx*	*Emacs*	*HTML*	*Sponsor*	*Status*
<HR> Ⓑ	✔*	✔	✔	✔	✔*	✔	2	W3C	
align=	✔*	✔	✔	✔	✔*	✔	3	W3C	
(center, left, right)									
(justify)							3	W3C	✗3+
class=	-	-	-	-	-	-	4	W3C	
clear=	-	-	-	-	-	-	3	W3C	✗3+
(left, right, all)									
color=	4	-	-	-	-	-	Ⓟ	MS	✗3+
id=	4	-	-	-	-	-	3/4	W3C	
md=	-	-	-	-	-	-	3	W3C	✗3+
noshade=	✔	✔	✔	✔	✔*	✔	3+/4	W3C	
nowrap	-	-	-	-	-	-	3	W3C	✗3+
size=	4	✔	-	✔	✔	✔	3+	W3C	↓4
src=	4	-	-	-	-	-	3	W3C	✗3+
style=	4	-	-	-	-	-	4	W3C	
title=	4	-	-	-	-	-	4	W3C	
width=	✔	✔	✔	✔	✔	✔	3+	W3C	↓4
event=	4	-	-	-	-	-	4	W3C	

Etichetele și atributele care sunt scrise cu litere **aldine** sunt incluse în limbajul HTML 4.0, iar etichetele între paranteze drepte sunt opționale. Tabelul este aranjat în ordine alfabetică, pentru o regăsire mai ușoară, dar tipurile Structure/Block/Text ale etichetelor sunt identificate prin Ⓢ, Ⓑ, Ⓣ după eticheta propriu-zisă.

2,3,4 etc. = Numărul versiunii HTML condensate; 3+ = versiunea HTML 3.2; ✔ = disponibil pentru toate versiunile curente; ↓ nerecomandat – versiunea HTML; ✗ = depășit; Ⓟ = propriu/neconform standardului; *= implementare parțială sau parțial neconform standardului; NS=Non Specified.

Eticheta	*MSIE*	*Netscp*	*NCSA*	*Opera*	*Lynx*	*Emacs*	*HTML*	*Sponsor*	*Status*
<HTML>...[</HTML>] Ⓢ	✔	✔	✔	✔	✔	✔	2-4	W3C	
dir=	-	-	-	-	-	-	4	W3C	
lang=	-	-	-	-	-	-	4	W3C	
title=	4	-	-	-	-	-			✘4
version=	-	-	-	-	-	-	4	W3C	
<I></I> Ⓣ	✔	✔	✔	✔	✔*	✔	2-4	W3C	
class=	-	-	-	-	✔	✔	3/4	W3C	
dir=	-	-	-	-	-	-	4	W3C	
id=	✔3/4	-	-	-	✔	✔	3/4	W3C	
lang=	-	-	-	-	✔	✔	3/4	W3C	
style=	4	-	-	-	-	-	4	W3C	
title=	4	-	-	-	-	-	4	W3C	
event=	4	-	-	-	-	-	4	W3C	
<IFRAME></IFRAME> Ⓢ	✔3/4	✔	-	-	-	-	4	W3C	
align=	✔3/4	-	-	-	-	-	4	W3C	
(absbottom, absmiddle, baseline, bottom, left, middle, right, texttop, top)									
border=	4	✔	-	-	-	-			✘4
bordercolor=	4	-	-	-	-	-			✘4
frameborder=	✔	4	-	-	-	-	4	W3C	
framespacing=	4	-	-	-	-	-			✘4
height=	4	-	-	-	-	-		W3C	✘4
hspace=	4	-	-	-	-	-		W3C	✘4
id=	4	-	-	-	-	-		W3C	✘4
marginheight=	4	-	-	-	-	-	4	W3C	

Eticheta	*MSIE*	*Netscp*	*NCSA*	*Opera*	*Lynx*	*Emacs*	*HTML*	*Sponsor*	*Status*
marginwidth=	4	-	-	-	-	-		W3C	
name=	4	✔	-	-	-	-	4	W3C	
(window_name. _blank, _parent, _self, _top)									
noresize=	✔	4	-	-	-	-	4	W3C	
(noresize, resize)									
scrolling=	4	-	-	-	-	-	4	W3C	
(auto, no, yes)									
src=	4	-	-	-	-	-	4	W3C	
style=	4	-	-	-	-	-			✘4
title=	4	-	-	-	-	-			✘4
vspace=	4	-	-	-	-	-			✘4
width=	4	-	-	-	-	-	4	W3C	
event=	4	-	-	-	-	-	4	W3C	
<ILAYER></ILAYER> Ⓢ	-	4	-	-	-	-	Ⓟ	NS	
OBSERVAȚIE: Eticheta <ILAYER> este similară cu eticheta <LAYER>									
<IMG> Ⓑ	✔	✔	✔	✔	✔*	✔*	2	W3C	
OBSERVAȚIE: Browserele în mod text, cum este Lynx, redă întotdeauna atributul `alt=`, deși ele ar putea (sau nu) să facă imaginea disponibilă pentru redare cu un utilitar separat. În general, este cel mai bine să includeți întotdeauna o valoare `ALT=`, stabilind-o explicit la valoarea `null` pentru imagini pur decorative care nu ar trebui să fie etichetate.									
align=	✔	✔	✔*	✔	✔*	✔	2	W3C	

Etichetele și atributele care sunt scrise cu litere **aldine** sunt incluse în limbajul HTML 4.0, iar etichetele între paranteze drepte sunt opționale. Tabelul este aranjat în ordine alfabetică, pentru o regăsire mai ușoară, dar tipurile Structure/Block/Text ale etichetelor sunt identificate prin Ⓢ, Ⓑ, Ⓣ după eticheta propriu-zisă.

2,3,4 etc. = Numărul versiunii HTML condensate; 3+ = versiunea HTML 3.2; ✔ = disponibil pentru toate versiunile curente; ↓ nerecomandat – versiunea HTML; ✘ = depășit; Ⓟ = propriu/neconform standardului; *= implementare parțială sau parțial neconform standardului; NS=Non Specified.

Eticheta	*MSIE*	*Netscp*	*NCSA*	*Opera*	*Lynx*	*Emacs*	*HTML*	*Sponsor*	*Status*
<IMG> *continuare*									
(bottom, left, middle, right, top)							2	W3C	
(absbottom, absmiddle, baseline, texttop)									
alt=	✔	✔	✔	✔	✔	✔	2-4	W3C	
border=	✔	✔	✔	✔	-	-	3+/4	W3C	
class=	-	-	-	-	-	-	3/4	W3C	
controls=	✔	-	-	-	-	-	Ⓟ		✘4
dir=	-	-	-	-	-	-	4	W3C	
datafld=	4	-	-	-	-	-	Ⓟ		✘4
datasrc=	4	-	-	-	-	-	Ⓟ		✘4
dynsrc=	4	-	-	-	-	-	Ⓟ		✘4
height=	✔	✔	-	✔	-	✔*	2-4	W3C	
hspace=	✔3/4	✔	-	✔	-	-	3+/4	W3C	
id=	✔3/4	-	-	-	-	-	3/4	W3C	
ismap=	✔	✔	✔	✔	✔*	✔*	2-4	W3C	
lang=	-	-	-	-	-	-	3/4	W3C	
loop=	4	-	-	-	-	-	Ⓟ		✘4
lowsrc=	4	✔	-	-	-	-	Ⓟ		✘4
md=	-	-	-	-	-	-	3	W3C	✘3+
name=	4	-	-	-	-	-			
src=	✔	✔	✔	✔	✔*	✔*	2-4	W3C	
style=	4	-	-	-	-	-	4	W3C	
title=	4	-	-	-	-	-	4	W3C	
units=	-	-	-	-	-	-	3	W3C	✘3+
(pixels, en)									

Eticheta	*MSIE*	*Netscp*	*NCSA*	*Opera*	*Lynx*	*Emacs*	*HTML*	*Sponsor*	*Status*
usemap=	✔	✔	-	✔	✔*	✔*	3+/4	W3C	
vrml=	4	-	-	-	-	-			✘4
vspace=	✔	✔	-	✔	-	-	3+/4	W3C	
width=	✔	✔	-	✔	-	✔	2	W3C	
event=	4	-	-	-	-	-	4	W3C	
<INPUT>[</INPUT>] Ⓢ	✔	✔	✔	✔	✔*	✔	2-4	W3C	
accesskey=	4	-	-	-	-	-	Ⓟ	MS	
accept=	-	-	-	-	-	-	4	W3C	
align=	✔	✔	-	✔	-	-	3/4	W3C	
(bottom, left, middle, right, top)									
(absbottom, absmiddle, baseline, texttop									
alt=	-	-	-	-	-	-	4	W3C	
checked=	✔	✔	✔	✔	✔	✔	3/4	W3C	
class=	-	-	-	-	-	-	3/4	W3C	
datafld=	4	-	-	-	-	-	Ⓟ		✘4
datasrc=	4	-	-	-	-	-*	Ⓟ		✘4
dir=	-	-	-	-	-	-	4	W3C	
disabled	4	-	-	-	-	-	3/4	W3C	
error	-	-	-	-	-	-	3	W3C	✘3+

Etichetele și atributele care sunt scrise cu litere **aldine** sunt incluse în limbajul HTML 4.0, iar etichetele între paranteze drepte sunt opționale. Tabelul este aranjat în ordine alfabetică, pentru o regăsire mai ușoară, dar tipurile Structure/Block/Text ale etichetelor sunt identificate prin Ⓢ, Ⓑ, Ⓣ după eticheta propriu-zisă.

2,3,4 etc. = Numărul versiunii HTML condensate; 3+ = versiunea HTML 3.2; ✔ = disponibil pentru toate versiunile curente; ↓ nerecomandat – versiunea HTML; ✘ = depășit; Ⓟ = propriu/neconform standardului; *= implementare parțială sau parțial neconform standardului; NS=Non Specified.

Eticheta	*MSIE*	*Netscp*	*NCSA*	*Opera*	*Lynx*	*Emacs*	*HTML*	*Sponsor*	*Status*
<INPUT>[</INPUT>] *continuare*									
id=	4	-	-	-	-	-	3/4	W3C	
lang=	-	-	-	-	-	-	3/4	W3C	
language=	4	-	-	-	-	-			✗4
(javascript, vbscript)									
max=	✔	✔	-	-	-	-	3	W3C	✗3+
maxlength=	✔	✔	✔	✔	✔	✔	2-4	W3C	
md=	-	-	-	-	-	-	3	W3C	✗3+
min=	✔	✔	-	-	-	-	3	W3C	✗3+
name=	✔	✔	✔	✔	✔	✔	2-4	W3C	
readonly=	4	-	-	-	-	-	4	W3C	
size=	✔	✔	✔	✔	✔	✔	2-4	W3C	
src=	-	-	-	-	-	-	2-4	W3C	
style=	4	-	-	-	-	-	4	W3C	
tabindex=	4	4*	-	-	-	-	4	W3C	
title=	4	-	-	-	-	-	4	W3C	
type=	✔	✔	✔	✔	✔*	✔	2-4	W3C	
(button, checkbox, hidden, image, password, radio, reset, select-multiple, select-one, submit, text, textarea)									
usemap=	-	-	-	-	-	-	4	W3C	
value=	✔	✔	✔	✔	✔	✔	2	W3C	✗4
event=	4	-	-	-	-	-	4		
<INS></INS> Ⓣ	-	-	-	-	✔*	✔	3/4	W3C	✗3+
cite=	-	-	-	-	-	-	4	W3C	
class=	-	-	-	-	✔	✔	3/4	W3C	

Eticheta	*MSIE*	*Netscp*	*NCSA*	*Opera*	*Lynx*	*Emacs*	*HTML*	*Sponsor*	*Status*
datetime=	-	-	-	-	-	-	4	W3C	
dir=	-	-	-	-	-	-		W3C	
id=	-	-	-	-	-	-	3	W3C	
lang=	-	-	-	-	-	-	3	W3C	
style=	-	-	-	-	-	-	-	W3C	
title=	-	-	-	-	-	-	-	W3C	
event=	-	-	-	-	-	-	-	W3C	
<ISINDEX> Ⓑ	✔	✔	✔	✔	✔	✔	2-4	W3C	↓4
OBSERVAȚIE: Folosiți, dacă este posibil, eticheta <FORM> în locul etichetei <ISINDEX>									
action=	-	-	-	-	-	-	-		✗3+
class=	-	-	-	-	-	-	4	W3C	
dir=	-	-	-	-	-	-	4	W3C	
id=	-	-	-	-	-	-	4	W3C	
lang=	-	-	-	-	-	-	4	W3C	
prompt=	✔	✔	-	✔	-	-	4	W3C	
style=	-	-	-	-	-	-	4	W3C	
title=	-	-	-	-	-	-	4	W3C	
<KBD></KBD> Ⓣ	✔	✔	✔	✔	✔*	✔	2-4	W3C	
class=	-	-	-	-	✔	✔	3/4	W3C	

Etichetele și atributele care sunt scrise cu litere **aldine** sunt incluse în limbajul HTML 4.0, iar etichetele între paranteze drepte sunt opționale. Tabelul este aranjat în ordine alfabetică, pentru o regăsire mai ușoară, dar tipurile Structure/Block/Text ale etichetelor sunt identificate prin Ⓢ, Ⓑ, Ⓣ după eticheta propriu-zisă.

2,3,4 etc. = Numărul versiunii HTML condensate; 3+ = versiunea HTML 3.2; ✔ = disponibil pentru toate versiunile curente; ↓ nerecomandat – versiunea HTML; ✗ = depășit; Ⓟ = propriu/neconform standardului; *= implementare parțială sau parțial neconform standardului; NS=Non Specified.

Eticheta	*MSIE*	*Netscp*	*NCSA*	*Opera*	*Lynx*	*Emacs*	*HTML*	*Sponsor*	*Status*
<KBD></KBD> *continuare*									
dir=	-	-	-	-	-	-	4	W3C	
id=	✔3/4	-	-	-	✔	✔	3/4	W3C	
lang=	-	-	-	-	✔	✔	3/4	W3C	
style=	4	-	-	-	-	-	4	W3C	
title=	4	-	-	-	-	-	4	W3C	
event=	4	-	-	-	-	-	4	W3C	
<KEYGEN></KEYGEN> Ⓢ	-	✔	-	-	-	-	Ⓟ	NS	
<LABEL></LABEL> Ⓣ	✔3/4	-	-	-	-	-	4	W3C	
accesskey=	4	-	-	-	-	-	4	W3C	
class=	-	-	-	-	-	-	4	W3C	
dir=	-	-	-	-	-	-	4	W3C	
disabled=	-	-	-	-	-	-	4	W3C	
for=	4	-	-	-	-	-	4	W3C	
id=	4	-	-	-	-	-	4	W3C	
lang=	-	-	-	-	-	-	4	W3C	
style=	4	-	-	-	-	-	4	W3C	
title=	4	-	-	-	-	-	4	W3C	
event=	4	-	-	-	-	-	4	W3C	
<LAYER></LAYER> Ⓢ	-	4	-	-	-	-	Ⓟ	NS	
id=									
left=									
top=									
pagex=									

Eticheta	*MSIE*	*Netscp*	*NCSA*	*Opera*	*Lynx*	*Emacs*	*HTML*	*Sponsor*	*Status*
pagey=									
src=									
z-index=									
above=									
below=									
width=									
height=									
clip=									
visibility=									
(show, hidden, inherit)									
bgcolor=									
background=									
onmouseover=									
onmouseout=									
onfocus=									
onblur=									
onload=									
<LEGEND> Ⓢ	4	-	-	-	-	-	4	W3C	
accesskey=	4	-	-	-	-	-	4	W3C	

Etichetele și atributele care sunt scrise cu litere **aldine** sunt incluse în limbajul HTML 4.0, iar etichetele între paranteze drepte sunt opționale. Tabelul este aranjat în ordine alfabetică, pentru o regăsire mai ușoară, dar tipurile Structure/Block/Text ale etichetelor sunt identificate prin Ⓢ, Ⓑ, Ⓣ după eticheta propriu-zisă.

2,3,4 etc. = Numărul versiunii HTML condensate; 3+ = versiunea HTML 3.2; ✔ = disponibil pentru toate versiunile curente; ↓ nerecomandat – versiunea HTML; ✘ = depășit; Ⓟ = propriu/neconform standardului; *= implementare parțială sau parțial neconform standardului; NS=Non Specified.

Eticheta	*MSIE*	*Netscp*	*NCSA*	*Opera*	*Lynx*	*Emacs*	*HTML*	*Sponsor*	*Status*
<LEGEND> *continuare*									
align=	-	-	-	-	-	-	4	W3C	
class=	-	-	-	-	-	-	4	W3C	
dir=	-	-	-	-	-	-	4	W3C	
id=	4	-	-	-	-	-	4	W3C	
lang=	-	-	-	-	-	-	4	W3C	
style=	4	-	-	-	-	-	4	W3C	
title=	4	-	-	-	-	-	4	W3C	
event=	4	-	-	-	-	-	4	W3C	
<LI>[</LI>] Ⓢ	✔	✔	✔	✔	✔	✔	2	W3C	
align=	4	-	-	-	-	-			
(center, left, right)									
class=	-	-	-	-	-	-	4	W3C	
dir=	-	-	-	-	-	-	4	W3C	
id=	4	-	-	-	-	-	4	W3C	
lang=	-	-	-	-	-	-	4	W3C	
style=	4	-	-	-	-	-	4	W3C	
title=	4	-	-	-	-	-	4	W3C	
type=	✔	✔	✔	✔	✔	✔	3+/4	W3C	
(1, a, A, i, I)									
value=	4	-	-	-	-	-	3+/4	W3C	
event=	4	-	-	-	-	-	4	W3C	
<LINK>[</LINK>] Ⓢ	✔	✔	✔	✔	✔	✔	2-4	W3C	
	Ⓟ	-	-	-	-	-			
class=	-	-	-	-	-	-	4	W3C	

Eticheta	*MSIE*	*Netscp*	*NCSA*	*Opera*	*Lynx*	*Emacs*	*HTML*	*Sponsor*	*Status*
dir=	-	-	-	-	-	-	4	W3C	
href=	✔	✔	✔	✔	✔	✔	2-4	W3C	
id=	-	-	-	-	-	-	4	W3C	
lang=	-	-	-	-	-	-	4	W3C	
media=	-	-	-	-	-	-	4	W3C	
methods=	✔	✔	✔	✔	✔	✔	2	W3C	✘3+
name=	✔	-	-	-	-	-	2	W3C	✘3+
rel=	✔	✔	✔	✔	✔	✔	2-4	W3C	
(contents, home, toc, index, glossary, copyright, up, next,									
previous, start, help, bookmark, banner, stylesheet, alternate)									
rev=	✔	✔	✔	✔	✔	✔	2-4	W3C	
style=	-	-	-	-	-	-	4	W3C	
target=	✔	-	-	-	-	-	4	W3C	
title=	✔	✔	✔	✔	✔	✔	2-4	W3C	
type=	✔	-	-	-	-	-	4	W3C	
urn=	✔	✔	✔	✔	✔	✔	2	W3C	✘3+
<LISTING></LISTING> Ⓣ	✔	✔	✔	✔	✔	✔	2	W3C	↓2✘4
align=	4	-	-	-	-	-			
(center, left, right)									

Etichetele și atributele care sunt scrise cu litere **aldine** sunt incluse în limbajul HTML 4.0, iar etichetele între paranteze drepte sunt opționale. Tabelul este aranjat în ordine alfabetică, pentru o regăsire mai ușoară, dar tipurile Structure/Block/Text ale etichetelor sunt identificate prin Ⓢ, Ⓑ, Ⓣ după eticheta propriu-zisă.

2,3,4 etc. = Numărul versiunii HTML condensate; 3+ = versiunea HTML 3.2; ✔ = disponibil pentru toate versiunile curente; ↓ nerecomandat – versiunea HTML; ✘ = depășit; Ⓟ = propriu/neconform standardului; *= implementare parțială sau parțial neconform standardului; NS=Non Specified

Eticheta	*MSIE*	*Netscp*	*NCSA*	*Opera*	*Lynx*	*Emacs*	*HTML*	*Sponsor*	*Status*
<LISTING></LISTING> continuare									
class=	-	-	-	-	✔	✔	3	W3C	
id=	✔3/4	-	-	-	✔	✔	3	W3C	
lang=	-	-	-	-	-	-	3	W3C	
style=	4	-	-	-	-	-		W3C	
title=	4	-	-	-	-	-		W3C	
event=	4	-	-	-	-	-		W3C	
<MAP></MAP> Ⓢ	✔3/4	✔	✔	✔	✔*	✔*	3+/4	W3C	
SUGESTIE: Browserul Lynx de fapt nu afișează imaginea, el creează o listă de referințe a tuturor destinațiilor într-o hartă imagine pe partea clientului. Dacă numele destinațiilor au sens, utilizatorii fără acces la hartă pot naviga totuși folosind modul text analog construit.									
class=	-	-	-	-	-	-	4	W3C	
id=	4	-	-	-	-	-	4	W3C	
name=	✔	✔	-	-	✔	✔	3+/4	W3C	
style=	4	-	-	-	-	-	4	W3C	
title=	4	-	-	-	-	-	4	W3C	
event=	4	-	-	-	-	-	4	W3C	
<MARQUEE>...</MARQUEE>Ⓑ	✔	-	-	-	✔*	-	Ⓟ	MS	✔
OBSERVAȚIE: Browserul Lynx tratează eticheta Microsoft <MARQUEE> ca un sinonim pentru eticheta <BANNER> și o redă static (ca text simplu). În practică, majoritatea browserelor ignoră eticheta și redau conținutul ca text simplu.									
align=	✔	-	-	-	-	-			
(absbottom, absmiddle, baseline, bottom, left, middle, right, texttop, top)									
behavior=	✔	-	-	-	-	-			
(alternate, scroll, slide)									
bgcolor=	✔	-	-	-	-	-			
datafld=	4	-	-	-	-	-			

Eticheta	*MSIE*	*Netscp*	*NCSA*	*Opera*	*Lynx*	*Emacs*	*HTML*	*Sponsor*	*Status*
dataformats=	4	-	-	-	-	-			
datasrc=	4	-	-	-	-	-			
direction=	✔	-	-	-	-	-			
(down, left, right, up)									
height=	✔	-	-	-	-	-			
hspace=	✔	-	-	-	-	-			
id=	4	-	-	-	-	-			
loop=	✔	-	-	-	-	-			
scrollamount=	4	-	-	-	-	-			
scrolldelay=	4	-	-	-	-	-			
style=	4	-	-	-	-	-			
title=	4	-	-	-	-	-			
vspace=	✔	-	-	-	-	-			
width=	✔	-	-	-	-	-			
event=	4	-	-	-	-	-			
$$ Ⓣ	-	-	-	-	-	-	3	W3C	✘3+
OBSERVAȚIE: Numeroasele elemente matematice din limbajul HTML 3.0 nu sunt incluse în acest tabel deoarece nu au fost folosite în realitate niciodată și s-a renunțat la ele începând cu versiunea HTML 3.2. W3C lucrează la un nou standard de marcare matematică.									
<MENU></MENU> Ⓢ	✔	✔	✔	✔	✔	✔	2-4	W3C	↓4
compact	✔	✔	✔	✔	✔*	✔	2-4	W3C	↓4

Etichetele și atributele care sunt scrise cu litere **aldine** sunt incluse în limbajul HTML 4.0, iar etichetele între paranteze drepte sunt opționale. Tabelul este aranjat în ordine alfabetică, pentru o regăsire mai ușoară, dar tipurile Structure/Block/Text ale etichetelor sunt identificate prin Ⓢ, Ⓑ, Ⓣ după eticheta propriu-zisă.

2,3,4 etc. = Numărul versiunii HTML condensate; 3+ = versiunea HTML 3.2; ✔ = disponibil pentru toate versiunile curente; ↓ nerecomandat – versiunea HTML; ✘ = depășit; Ⓟ = propriu/neconform standardului; *= implementare parțială sau parțial neconform standardului; NS=Non Specified.

Eticheta	*MSIE*	*Netscp*	*NCSA*	*Opera*	*Lynx*	*Emacs*	*HTML*	*Sponsor*	*Status*
<MENU></MENU> *continuare*									
dir=	-	-	-	-	-	-	4	W3C	
id=	4	-	-	-	-	-	4	W3C	
lang=	-	-	-	-	-	-	4	W3C	
style=	4	-	-	-	-	-	4	W3C	
title=	4	-	-	-	-	-	4	W3C	
event=	4	-	-	-	-	-	4	W3C	
<META></META> Ⓢ	✔ Ⓟ	✔	✔	✔	✔	✔	2-4	W3C	
content=	✔	✔	✔	✔	✔	✔	2-4	W3C	
dir=	-	-	-	-	-	-	4	W3C	
http-equiv=	✔	✔	✔	✔	✔	✔	2-4	W3C	
lang=	-	-	-	-	-	-	4	W3C	
name=	✔	✔	✔	✔	✔	✔	2-4	W3C	
SUGESTIE: Atributele `NAME` și `HTTP-EQUIV` nu ar trebui să fie folosite împreună deoarece `NAME` se presupune că este identic cu `HTTP-EQUIV`. `HTTP-EQUIV` ar trebui să evite numele care corespund anteturilor HTTP interne, deoarece aceste valori sunt admise de către HTTP și sunt de obicei ignorate dacă se suprapun peste „cuvintele rezervate" HTTP.									
scheme=	-	-	-	-	-	-	4	W3C	
title=	4	-	-	-	-	-		W3C	✘4
url=	4	-	-	-	-	-			✘4
<MULTICOL> Ⓢ	-	✔	-	-	-	-	Ⓟ	NS	
<NEXTID> Ⓢ	✔	✔	✔	✔	✔	✔	2	W3C	✘3+
OBSERVAȚIE: Se pare că nimeni nu mai face nimic cu eticheta `<NEXTID>` și ea a dispărut din standard.									
n=	✔	✔	✔	✔	✔	✔	2	W3C	
(znnn)									

Eticheta	*MSIE*	*Netscp*	*NCSA*	*Opera*	*Lynx*	*Emacs*	*HTML*	*Sponsor*	*Status*
<NOBR>[<NOBR>] Ⓢ	✔	✔	-	✔	-	✔		NS	
id=	4	-	-	-	-	-	4	W3C	
style=	4	-	-	-	-	-	4	W3C	
title=	4	-	-	-	-	-	4	W3C	
<NOEMBED> Ⓑ	-	✔	-	-	-	-	Ⓟ	NS	
<NOFRAMES> Ⓢ	✔ 3/4	✔2-4	-	✔	✔*	✔	4	W3C	
...</NOFRAMES>									
id=	4	✔	-	-	-	-	4	MS	
style=	4	-	-	-	-	-	4	MS	
title=	4	-	-	-	-	-	4	MS	
<NOLAYER> Ⓢ	-	4	-	-	-	-	Ⓟ	NS	
...</NOLAYER>									
<NOSCRIPT> Ⓢ	✔ 3/4	✔3/4	-	✔	✔	✔		W3C	
...</NOSCRIPT>									
title=	4	-	-	-	-	-	4	W3C	
<NOTE></NOTE> Ⓢ	-	-	-	-	✔*	✔	3	W3C	✘3+
class=	-	-	-	-	✔*	✔	3	W3C	
clear=	-	-	-	-	✔*	✔	3	W3C	
(left, right, all)									

Etichetele și atributele care sunt scrise cu litere **aldine** sunt incluse în limbajul HTML 4.0, iar etichetele între paranteze drepte sunt opționale. Tabelul este aranjat în ordine alfabetică, pentru o regăsire mai ușoară, dar tipurile Structure/Block/Text ale etichetelor sunt identificate prin Ⓢ, Ⓑ, Ⓣ după eticheta propriu-zisă.

2,3,4 etc. = Numărul versiunii HTML condensate; 3+ = versiunea HTML 3.2; ✔ = disponibil pentru toate versiunile curente; ↓ nerecomandat – versiunea HTML; ✘ = depășit; Ⓟ = propriu/neconform standardului; *= implementare parțială sau parțial neconform standardului; NS=Non Specified.

Eticheta	*MSIE*	*Netscp*	*NCSA*	*Opera*	*Lynx*	*Emacs*	*HTML*	*Sponsor*	*Status*
<NOTE></NOTE> *continuare*									
id=	-	-	-	-	✔	✔	3	W3C	
lang=	-	-	-	-	✔	✔	3	W3C	
md=	-	-	-	-	-	-	3	W3C	
src=	-	-	-	-	✔*	✔	3	W3C	
<OBJECT></OBJECT> Ⓑ	✔ 3/4	-	-	-	✔	✔	4	W3C	
accesskey=	4	-	-	-	-	-	4	MS	
align=	✔ 3/4	-	-	-	-	-	4	W3C	↓4
(absbottom, absmiddle, baseline, bottom, left, middle, right, texttop, top)									
border=	✔3/4	-	-	-	-	-	4	W3C	
class=	✔3/4	-	-	-	-	-	4	W3C	
classid=	4	-	-	-	-	-	4	W3C	
code=	4	-	-	-	-	-			✘4
codebase=	4	-	-	-	-	-	4	W3C	
codetype=	4	-	-	-	-	-	4	W3C	
data=	✔3/4	-	-	-	-	-	4	W3C	
datafld=	4	-	-	-	-	-			✘4
datasrc=	4	-	-	-	-	-			✘4
declare=	-	-	-	-	-	-	4	W3C	
dir=	-	-	-	-	-	-	4	W3C	
disabled=	4	-	-	-	-	-			✘4
height=	✔3/4	-	-	-	-	-	4	W3C	
hspace=	-	-	-	-	-	-	4	W3C	
id=	4	-	-	-	-	-	4	W3C	
lang=	-	-	-	-	-	-	4	W3C	

Eticheta	*MSIE*	*Netscp*	*NCSA*	*Opera*	*Lynx*	*Emacs*	*HTML*	*Sponsor*	*Status*
name=	✔3/4	-	-	-	-	-	4	W3C	
shapes=	-	-	-	-	-	-	4	W3C	
standby=	-	-	-	-	-	-	4	W3C	
style=	4	-	-	-	-	-	4	W3C	
tabindex=	4	-	-	-	-	-	4	W3C	
title=	4	-	-	-	-	-	4	W3C	
type=	✔3/4	-	-	-	-	-	4	W3C	
usemap=	-	-	-	-	-	-	4	W3C	
vspace=	-	-	-	-	-	-	4	W3C	
width=	✔3/4	-	-	-	-	-	4	W3C	
event=	4	-	-	-	-	-	4	W3C	
<OL></OL> Ⓢ	✔	✔	✔*	✔	✔	✔	2-4	W3C	
align=	4	-	-	-	-	-			
(center, left, right)									
class=	-	-	-	-	✔	✔	3/4	W3C	
clear=	-	-	-	-	✔*	✔	3	W3C	✘3+
(left, right, all)									
compact=	-	-	-	✔	✔*	✔	2	W3C	↓4
continue	-	-	-	-	✔	✔	3	W3C	✘3+

Etichetele și atributele care sunt scrise cu litere **aldine** sunt incluse în limbajul HTML 4.0, iar etichetele între paranteze drepte sunt opționale. Tabelul este aranjat în ordine alfabetică, pentru o regăsire mai ușoară, dar tipurile Structure/Block/Text ale etichetelor sunt identificate prin Ⓢ, Ⓑ, Ⓣ după eticheta propriu-zisă.

2,3,4 etc. = Numărul versiunii HTML condensate; 3+ = versiunea HTML 3.2; ✔ = disponibil pentru toate versiunile curente; ↓ nerecomandat – versiunea HTML; ✘ = depășit; Ⓟ = propriu/neconform standardului; *= implementare parțială sau parțial neconform standardului; NS=Non Specified.

Eticheta	*MSIE*	*Netscp*	*NCSA*	*Opera*	*Lynx*	*Emacs*	*HTML*	*Sponsor*	*Status*
<OL></OL> *continuare*									
dir=	-	-	-	-	-	-	4	W3C	
id=	4	-	-	-	✔	✔	3/4	W3C	
lang=	-	-	-	-	✔	✔	3/4	W3C	
seqnum=	✔	✔	-	✔	✔	✔	3	W3C	✘3+
start=	✔3/4	✔	-	✔	✔	✔	3/4	W3C	
style=	4	-	-	-	-	-	4	W3C	
title=	4	-	-	-	-	-	4	W3C	
type=	✔	✔	-	✔	✔	✔	2-4	W3C	
(1, a, A, i, I)									
event=	4	-	-	-	-	-	4	W3C	
<OPTION> Ⓢ	✔	✔	✔	✔	✔*	✔	2-4	W3C	
...[</OPTION>]									
class=	-	-	-	-	-	-	4	W3C	
dir=	-	-	-	-	-	-	4	W3C	
disabled	-	-	-	-	-	-	4	W3C	
id=	4	-	-	-	-	-	4	W3C	
lang=	-	-	-	-	-	-	4	W3C	
name=	✔	✔	✔	✔	✔	✔	2-4	W3C	
selected=	✔	✔	✔	✔	✔*	✔	2-4	W3C	
style=	-	-	-	-	-	-	4	W3C	
title=	4	-	-	-	-	-	4	W3C	
value=	✔	✔	✔	✔	✔*	✔	2-4	W3C	
<OVERLAY>	-	-	-	-	✔*	✔	3	W3C	✘3+

OBSERVAȚIE: Eticheta `<OVERLAY>` făcea parte din specificația pentru figură `<FIG>` în HTML 3.0 și nu mai este folosită.

Eticheta	*MSIE*	*Netscp*	*NCSA*	*Opera*	*Lynx*	*Emacs*	*HTML*	*Sponsor*	*Status*
<P>[</P>] Ⓢ	✔	✔	✔	✔	✔	✔	2-4	W3C	

SUGESTIE: Unele browsere mai vechi inserează o linie suplimentară la vederea etichetei de încheiere `</P>`, așa că este mai bine să renunțați la ea, mai ales că standardul o prezintă ca fiind opțională. Această comportare este neconformă când este întâlnită, dar contează prea puțin deoarece majoritatea autorilor de HTML ignoră eticheta de încheiere.

align=	✔	✔	✔*	✔	✔	✔	3+	W3C	↓4
(center, left, right)							3	W3C	✘3+
(justify)							3/4	W3C	
class=	-	-	-	-	-	-	3	W3C	✘3+
clear=	-	✔	-	-	-	-	4	W3C	
(left, right, all)							3/4	W3C	
dir=	-	-	-	-	-	-	3/4	W3C	
id=	4	-	-	-	-	-	4	W3C	
lang=	-	-	-	-	-	-	4	W3C	
style=	4	-	-	-	-	-	2	W3C	✘3+
title=	4	-	-	-	-	-	4	W3C	
width=	4	-	-	-	-	-	3+	W3C	
event=	4	-	-	-	-	-			
<PARAM>[</PARAM>] Ⓑ	✔3/4	✔	-	-	-	-			
data=	4	-	-	-	-	-			
datafld=	4	-	-	-	-	-			

Etichetele și atributele care sunt scrise cu litere **aldine** sunt incluse în limbajul HTML 4.0, iar etichetele între paranteze drepte sunt opționale. Tabelul este aranjat în ordine alfabetică, pentru o regăsire mai ușoară, dar tipurile Structure/Block/Text ale etichetelor sunt identificate prin Ⓢ, Ⓑ, Ⓣ după eticheta propriu-zisă.

2,3,4 etc. = Numărul versiunii HTML condensate; 3+ = versiunea HTML 3.2; ✔ = disponibil pentru toate versiunile curente; ↓ nerecomandat – versiunea HTML; ✘ = depășit; Ⓟ = propriu/neconform standardului; *= implementare parțială sau parțial neconform standardului; NS=Non Specified.

Eticheta	*MSIE*	*Netscp*	*NCSA*	*Opera*	*Lynx*	*Emacs*	*HTML*	*Sponsor*	*Status*
<PARAM>[</PARAM>] *continuare*									
datasrc=	4	-	-	-	-	-			
name=	4	✔	-	-	-	-	3+/4	W3C	
object=	4	-	-	-	-	-			
ref=	4	-	-	-	-	-			
title=	4	-	-	-	-	-			
type=	-	-	-	-	-	-	4	W3C	
value=	4	✔	-	-	-	-	3+/4	W3C	
valuetype= (data, ref, object)	-	-	-	-	-	-	4	W3C	
<PERSON></PERSON> Ⓣ	-	-	-	-	✔*	✔	3	W3C	✘3+
class=	-	-	-	-	✔	✔	3	W3C	
id=	-	-	-	-	✔	✔	3	W3C	
lang=	-	-	-	-	✔	✔	3	W3C	
<PLAINTEXT> Ⓢ	✔	✔	✔	✔	✔	✔	2	W3C	↓2✘4
OBSERVAȚIE: Atât browserul MSIE cât și NCSA Mosaic au permis o etichetă de încheiere neconformă `</PLAINTEXT>`. Sintaxa ar fi trebuit să fie următoarea: `<HTML> ... <BODY> ... [</BODY>] [</HTML>] <PLAINTEXT> ...`									
id=	4	-	-	-	-	-		W3C	
style=	4	-	-	-	-	-		W3C	
title=	4	-	-	-	-	-		W3C	
event=	4	-	-	-	-	-		W3C	
<PRE></PRE> Ⓢ	✔	✔	✔	✔	✔	✔	2-4	W3C	
class=	-	-	-	-	-	-	3/4	W3C	

Eticheta	*MSIE*	*Netscp*	*NCSA*	*Opera*	*Lynx*	*Emacs*	*HTML*	*Sponsor*	*Status*
clear= (left, right, all)	-	-	-	-	-	-	3	W3C	✘3+
dir=	-	-	-	-	-	-		W3C	
id=	✔3/4	-	-	-	-	-	3/4	W3C	
lang=	-	-	-	-	-	-	3/4	W3C	
style=	4	-	-	-	-	-	4	W3C	
title=	4	-	-	-	-	-	4	W3C	
width=	4	-	-	-	-	-	2-4	W3C	
event=	4	-	-	-	-	-	4	W3C	
<Q></Q> Ⓣ	4	-	-	-	✔*	✔	3/4	W3C	
cite=	-	-	-	-	-	-	4	W3C	
class=	-	-	-	-	✔	✔	3/4	W3C	
dir=	-	-	-	-	-	-	4	W3C	
id=	✔3/4	-	-	-	✔	✔	3/4	W3C	
lang=	-	-	-	-	✔	✔	3/4	W3C	
style=	4	-	-	-	-	-	4	W3C	
title=	4	-	-	-	-	-	4	W3C	
event=	4	-	-	-	-	-	4	W3C	

Etichetele și atributele care sunt scrise cu litere **aldine** sunt incluse în limbajul HTML 4.0, iar etichetele între paranteze drepte sunt opționale. Tabelul este aranjat în ordine alfabetică, pentru o regăsire mai ușoară, dar tipurile Structure/Block/Text ale etichetelor sunt identificate prin Ⓢ, Ⓑ, Ⓣ după eticheta propriu-zisă.

2,3,4 etc. = Numărul versiunii HTML condensate; 3+ = versiunea HTML 3.2; ✔ = disponibil pentru toate versiunile curente; ↓ nerecomandat – versiunea HTML; ✘ = depășit; Ⓟ = propriu/neconform standardului; *= implementare parțială sau parțial neconform standardului; NS=Non Specified.

Eticheta	*MSIE*	*Netscp*	*NCSA*	*Opera*	*Lynx*	*Emacs*	*HTML*	*Sponsor*	*Status*
<RANGE>	-	-	-	-	-	-	3	W3C	✘3+
class=	-	-	-	-	-	-	3	W3C	
from=	-	-	-	-	-	-	3	W3C	
id=	-	-	-	-	-	-	3	W3C	
until=	-	-	-	-	-	-	3	W3C	
<S></S> Ⓣ	✔	✔	✔	✔	✔*	✔	2	W3C	✘3+↓4
class=	-	✔3/4	-	-	✔	✔	3/4	W3C	
dir=	-	-	-	-	-	-	4	W3C	
id=	✔3/4	✔3/4	-	-	✔	✔	3/4	W3C	
lang=	-	✔3/4	-	-	✔	✔	3/4	W3C	
style=	4	-	-	-	-	-	4	W3C	
title=	4	-	-	-	-	-	4	W3C	
event=	4	-	-	-	-	-	4	W3C	
<SAMP></SAMP> Ⓣ	✔	✔	✔	✔	✔*	✔	2-4	W3C	
class=	-	-	-	-	✔	✔	3/4	W3C	
dir=	-	-	-	-	-	-	4	W3C	
id=	✔3/4	-	-	-	✔	✔	3/4	W3C	
lang=	-	-	-	-	✔	✔	3/4	W3C	
style=	4	-	-	-	-	-	4	W3C	
title=	4	-	-	-	-	-	4	W3C	
event=	4	-	-	-	-	-	4	W3C	
<SCRIPT></SCRIPT> Ⓑ	✔*	✔	-	3	-	-	3+/4	W3C	

ATENȚIE: Implementarea Microsoft pentru JavaScript de la Netscape, JScript, a fost incompatibilă cu „standardul" Netscape. S-a spus că va fi rezolvată această situație nefericită, dar codul Microsoft JScript creează încă probleme pentru Netscape și invers, dacă nu se iau anumite măsuri pentru a remedia diferențele.

Eticheta	*MSIE*	*Netscp*	*NCSA*	*Opera*	*Lynx*	*Emacs*	*HTML*	*Sponsor*	*Status*
event=	4	-	-	-	-	-			✘4
for=	4	-	-	-	-	-			✘4
id=	4	-	-	-	-	-			✘4
in=	4	-	-	-	-	-			✘4
language= (javascript, vbscript)	✔*	✔	-	3	-	-	4	W3C	
library=	4	-	-	-	-	-			✘4
src=	✔	✔	-	-	-	-	4	W3C	
title=	4	-	-	-	-	-	4	W3C	
type=	-	-	-	-	-	-	4	W3C	
<SELECT></SELECT> Ⓢ	✔	✔	✔	✔	✔	✔	2-4	W3C	
accesskey=	4	-	-	-	-	-	Ⓟ	MS	
align= (bottom, left, middle, right, top, absbottom, absmiddle, baseline, texttop)	4	-	-	-	-	-	3	W3C	✘3+
class=	-	✔3/4	-	-	-	-	3/4	W3C	
datafld=	✔	-	-	-	-	-	Ⓟ		
datasrc=	-	-	-	-	-	-	Ⓟ		
dir=	-	-	-	-	-	-	4	W3C	
disabled	4	-	-	-	-	-	3/4	W3C	

Etichetele și atributele care sunt scrise cu litere **aldine** sunt incluse în limbajul HTML 4.0, iar etichetele între paranteze drepte sunt opționale. Tabelul este aranjat în ordine alfabetică, pentru o regăsire mai ușoară, dar tipurile Structure/Block/Text ale etichetelor sunt identificate prin Ⓢ, Ⓑ, Ⓣ după eticheta propriu-zisă.

2,3,4 etc. = Numărul versiunii HTML condensate; 3+ = versiunea HTML 3.2; ✔ = disponibil pentru toate versiunile curente; ↓ nerecomandat – versiunea HTML; ✘ = depășit; Ⓟ = propriu/neconform standardului; *= implementare parțială sau parțial neconform standardului; NS=Non Specified.

Eticheta	*MSIE*	*Netscp*	*NCSA*	*Opera*	*Lynx*	*Emacs*	*HTML*	*Sponsor*	*Status*
<SELECT></SELECT> *continuare*									
height=	-	-	-	-	-	-	3	W3C	✘3+
id=	4	✔3/4	-	-	-	-	4	W3C	
lang=	-	✔3/4	-	-	-	-	3/4	W3C	
language=	4	-	-	-	-	-			✘4
(javascript, vbscript)									
md=	-	-	-	-	-	-	3	W3C	✘3+
multiple=	✔	✔	-	✔	-	-	2-4	W3C	
name=	✔	✔	-	✔	-	-	2-4	W3C	
readonly=	4	-	-	-	-	-			✘4
size=	✔	✔	-	✔	-	-	2-4	W3C	
style=	4	4	-	-	-	-	4	W3C	
tabindex=	4	4*	-	-	-	-	4	W3C	
title=	4	-	-	-	-	-	4	W3C	
units=	-	-	-	-	-	-	3	W3C	✘3+
width=	-	-	-	-	-	-	3	W3C	✘3+
event=	4	-	-	-	-	-	4	W3C	
<SERVER>	-	✔	-	-	-	-	Ⓟ	NS	
<SMALL></SMALL> Ⓣ	✔	✔	-	✔	✔*	✔	3/4	W3C	
class=	-	3	-	-	✔	✔	3/4	W3C	
dir=	-	-	-	-	-	-	4	W3C	
id=	✔3/4	3	-	-	✔	✔	3/4	W3C	
lang=	-	3	-	-	✔	✔	3/4	W3C	
style=	4	4	-	-	-	-	4	W3C	

Eticheta	*MSIE*	*Netscp*	*NCSA*	*Opera*	*Lynx*	*Emacs*	*HTML*	*Sponsor*	*Status*
title=	4	-	-	-	-	-	4	W3C	
event=	4	-	-	-	-	-	4	W3C	
<SPACER> Ⓣ	-	✔3/4	-	-	-	-	Ⓟ	NS	
<SPAN></SPAN> Ⓣ	✔3/4	4	-	-	-	-	4	W3C	
align=	-	-	-	-	-	-	4	W3C	
class=	-	-	-	-	-	-	4	W3C	
datafld=	4	-	-	-	-	-	Ⓟ		✘4
dataformats=	4	-	-	-	-	-	Ⓟ		✘4
datasrc=	4	-	-	-	-	-	Ⓟ		✘4
dir=	-	-	-	-	-	-	4	W3C	
id=	4	3	-	-	-	-	4	W3C	
style=	4	4	-	-	-	-	4	W3C	
title=	4	-	-	-	-	-	4	W3C	
event=	4	-	-	-	-	-	4	W3C	
<SPOT> Ⓣ	-	-	-	-	-	-	3	W3C	✘3+
id=	-	-	-	-	-	-	3	W3C	
<STRIKE></STRIKE> Ⓢ	✔	✔	✔	✔	✔*	✔	3+/4	W3C	↓4
class=	-	✔3/4	-	-	✔	✔	4	W3C	
dir=	-	-	-	-	-	-	4	W3C	

Etichetele și atributele care sunt scrise cu litere **aldine** sunt incluse în limbajul HTML 4.0, iar etichetele între paranteze drepte sunt opționale. Tabelul este aranjat în ordine alfabetică, pentru o regăsire mai ușoară, dar tipurile Structure/Block/Text ale etichetelor sunt identificate prin Ⓢ, Ⓑ, Ⓣ după eticheta propriu-zisă.

2,3,4 etc. = Numărul versiunii HTML condensate; 3+ = versiunea HTML 3.2; ✔ = disponibil pentru toate versiunile curente; ↓ nerecomandat – versiunea HTML; ✘ = depășit; Ⓟ = propriu/neconform standardului; *= implementare parțială sau parțial neconform standardului; NS=Non Specified.

Eticheta	*MSIE*	*Netscp*	*NCSA*	*Opera*	*Lynx*	*Emacs*	*HTML*	*Sponsor*	*Status*
<STRIKE></STRIKE> *continuare*									
id=	4	✔3/4	-	-	✔	✔	4	W3C	
lang=	-	✔3/4	-	-	✔	✔	4	W3C	
style=	4	4	-	-	-	-	4	W3C	
title=	4	-	-	-	-	-	4	W3C	
event=	4	-	-	-	-	-	4	W3C	
<STRONG></STRONG> Ⓣ	✔	✔	✔	✔	✔*	✔	2-4	W3C	
class=	-	3	-	-	✔	✔	3/4	W3C	
dir=	-	-	-	-	-	-	4	W3C	
id=	✔3/4	3	-	-	✔	✔	3/4	W3C	
lang=	-	3	-	-	✔	✔	3/4	W3C	
style=	4	4	-	-	-	-	4	W3C	
title=	4	-	-	-	-	-	4	W3C	
event=	4	-	-	-	-	-	4	W3C	
<STYLE></STYLE> Ⓣ	✔3/4	4	-	✔	✔*	✔	3+/4	W3C	
dir=	-	-	-	-	-	-	4	W3C	
lang=	-	-	-	-	-	-	4	W3C	
media=	-	-	-	-	-	-	4	W3C	
title=	4	-	-	-	-	-	4	W3C	
type=	4	-	-	-	✔	-	4	W3C	
<SUB> Ⓣ	✔	✔	✔	✔	✔*	✔	3/4	W3C	
class=	-	3	-	-	✔	✔	3/4	W3C	
dir=	-	-	-	-	-	-	4	W3C	
id=	3/4	3	-	-	✔	✔	3/4	W3C	

Eticheta	*MSIE*	*Netscp*	*NCSA*	*Opera*	*Lynx*	*Emacs*	*HTML*	*Sponsor*	*Status*
lang=	-	3	-	-	✔	✔	3/4	W3C	
style=	4	4	-	-	-	-	4	W3C	
title=	4	-	-	-	-	-	4	W3C	
event=	4	-	-	-	-	-	4	W3C	
<SUP> Ⓣ	✔	✔	✔	✔	✔*	✔	3/4	W3C	
class=	-	3	-	-	✔	✔	3/4	W3C	
dir=	-	-	-	-	-	-	4	W3C	
id=	3/4	3	-	-	✔	✔	3/4	W3C	
lang=	-	3	-	-	✔	✔	3/4	W3C	
style=	4	4	-	-	-	-	4	W3C	
title=	4	-	-	-	-	-	4	W3C	
event=	4	-	-	-	-	-	4	W3C	
<TAB> Ⓑ Ⓣ	-	-	-	-	✔	-	3	W3C	✘4
align=	-	-	-	-	✔	-	3	W3C	✘4
dp=	-	-	-	-	✔	-	3	W3C	✘4
id=	-	-	-	-	✔	-	3	W3C	✘4
indent=	-	-	-	-	✔	-	3	W3C	✘4
to=	-	-	-	-	✔	-	3	W3C	✘4

Etichetele și atributele care sunt scrise cu litere **aldine** sunt incluse în limbajul HTML 4.0, iar etichetele între paranteze drepte sunt opționale. Tabelul este aranjat în ordine alfabetică, pentru o regăsire mai ușoară, dar tipurile Structure/Block/Text ale etichetelor sunt identificate prin Ⓢ, Ⓑ, Ⓣ după eticheta propriu-zisă.

2,3,4 etc. = Numărul versiunii HTML condensate; 3+ = versiunea HTML 3.2; ✔ = disponibil pentru toate versiunile curente; ↓ nerecomandat – versiunea HTML; ✘ = depășit; Ⓟ = propriu/neconform standardului; *= implementare parțială sau parțial neconform standardului; NS=Non Specified.

Eticheta	*MSIE*	*Netscp*	*NCSA*	*Opera*	*Lynx*	*Emacs*	*HTML*	*Sponsor*	*Status*
<TABLE></TABLE> Ⓢ	✔	✔	✔	✔	✔*	✔	3+/4	W3C	
align=	✔	✔	-	✔	✔*	✔	3+/4	W3C	
(left, center, right)									
(justify, bleedleft, bleedright)							3	W3C	✘3+
class=	-	3	-	-	-	-	3/4	W3C	✘3+
background=	✔3/4	4*	-	3	-	-			✘4
bgcolor=	✔3/4	4	-	3	-	-	4	W3C	
border=	✔	✔	✔	✔	✔*	✔	3+/4	W3C	
bordercolor=	✔3/4	-	-	-	-	-	Ⓟ	MS	✘4
bordercolordark=	✔3/4	-	-	-	-	-	Ⓟ	MS	✘4
bordercolorlight=	✔3/4	-	-	-	-	-	Ⓟ	MS	✘4
cellpadding=	4	✔	-	✔	-	-	3+/4	W3C	
cellspacing=	4	✔	-	✔	-	-	3+/4	W3C	
class=	-	✔3/4	-	-	-	-	3/4	W3C	
clear=	-	-	-	-	-	-	3	W3C	✘3+
(left, right, all)									
cols=	4	-	-	-	-	-	4	W3C	
colspec=	-	-	-	-	-	-	3	W3C	✘3+
datasrc=	4	-	-	-	-	-	Ⓟ		✘3+
dir=	-	-	-	-	-	-	4	W3C	
dp=	-	-	-	-	-	-	3	W3C	✘3+
frame=	4	-	-	-	-	-	4	W3C	
(above, below, border, box, insides, lhs, rhs, void, vsides)									
height=	4	-	-	-	-	-			✘4
id=	4	✔3/4	-	-	-	-	3/4	W3C	

Eticheta	*MSIE*	*Netscp*	*NCSA*	*Opera*	*Lynx*	*Emacs*	*HTML*	*Sponsor*	*Status*
lang=	-	✔3/4	-	-	-	-	3/4	W3C	
noflow	-	-	-	-	-	-	3	W3C	✘3+
nowrap	✔	✔	-	-	-	-	3	W3C	✘3+
rules=	4	-	-	-	-	-	4	W3C	
(all, cols, groups, none, rows)									
style=	4	4	-	-	-	-	4	W3C	
title=	4	-	-	-	-	-	4	W3C	
units=	-	-	-	-	-	-	3	W3C	✘3+
width=	4	-	-	-	-	-	3+/4	W3C	
event=	4	-	-	-	-	-	4	W3C	
<TBODY>...[</TBODY>] Ⓑ	✔3/4	4	-	✔	-	-	4	W3C	
align=	4	-	-	-	-	-	4	W3C	
(center, left, right)									
bgcolor=	4	-	-	-	-	-		W3C	
char=	-	-	-	-	-	-		W3C	
charoff=	-	-	-	-	-	-		W3C	
class=	-	-	-	-	-	-	4	W3C	
dir=	-	-	-	-	-	-	4	W3C	
id=	4	-	-	-	-	-	4	W3C	

Etichetele și atributele care sunt scrise cu litere **aldine** sunt incluse în limbajul HTML 4.0, iar etichetele între paranteze drepte sunt opționale. Tabelul este aranjat în ordine alfabetică, pentru o regăsire mai ușoară, dar tipurile Structure/Block/Text ale etichetelor sunt identificate prin Ⓢ, Ⓑ, Ⓣ după eticheta propriu-zisă.

2,3,4 etc. = Numărul versiunii HTML condensate; 3+ = versiunea HTML 3.2; ✔ = disponibil pentru toate versiunile curente; ↓ nerecomandat – versiunea HTML; ✘ = depășit; Ⓟ = propriu/neconform standardului; *= implementare parțială sau parțial neconform standardului; NS=Non Specified.

Eticheta	*MSIE*	*Netscp*	*NCSA*	*Opera*	*Lynx*	*Emacs*	*HTML*	*Sponsor*	*Status*
<TBODY>...[</TBODY>] *continuare*									
lang=	-	-	-	-	-	-	4	W3C	
style=	4	-	-	-	-	-	4	W3C	
title=	4	-	-	-	-	-	4	W3C	
valign=	4	-	-	-	-	-	4	W3C	
(baseline, bottom, center, top)									
event=	4	-	-	-	-	-	4	W3C	
<TD></TD> Ⓢ	✔	✔	✔	✔	✔*		+/4	W3C	
align=	✔3/4*	✔	✔	✔	-	-	3+/4	W3C	
(center, left, right)							3	W3C	✘3+
(decimal)							3	W3C	✘3+
(justify)							3/4	W3C	
axis=	-	-	-	-	-	-	3/4	W3C	
axes=	-	-	-	-	-	-			✘4
background=	4	-	-	-	-	-	4	W3C	
bgcolor=	4	4	-	3	-	-			✘4
bordercolor=	4	-	-	-	-	-		MS	✘4
bordercolordark=	4	-	-	-	-	-		MS	✘4
bordercolorlight=	4	-	-	-	-	-	4	W3C	
char=	-	-	-	-	-	-	4	W3C	
charoff=	-	-	-	-	-	-	4	W3C	
class=	-	-	-	-	-	-	3+/4	W3C	
colspan=	✔	✔	-	✔	-	-	4	W3C	
dir=	-	-	-	-	-	-	3	W3C	✘3+
dp=	-	-	-	-	-	-			

Eticheta	*MSIE*	*Netscp*	*NCSA*	*Opera*	*Lynx*	*Emacs*	*HTML*	*Sponsor*	*Status*
height=	4	-	-	-	-	-	3+	W3C	✘4
id=	4	3	-	-	-	-	3/4	W3C	
lang=	-	3	-	-	-	-	3/4	W3C	
nowrap	✔	✔	-	✔	-	-	3+	W3C	✘4
rowspan=	4	✔	✔	✔	-	-	3+	W3C	✘4
style=	4	4	-	-	-	-	4	W3C	
title=	4	-	-	-	-	-	4	W3C	
valign=	✔	✔	✔	✔	-	-	3+/4	W3C	
(baseline, **bottom, center, top**)									
width=	4	-	-	-	-	-	3+	W3C	✘4
event=	4	-	-	-	-	-	4	W3C	
<TEXTAREA> Ⓢ	✔	✔	✔	✔	✔*	✔	2-4	W3C	
...</TEXTAREA>							Ⓟ	MS	
acceskey=	4	-	-	-	-	-	3	W3C	✘4
align=	4	-	-	-	-	-			
(bottm, left, middle , right, top)									
(absbottom, absmiddle, baseline, texttop)									
class=	-	✔3/4	-	-	-	-	3/4	W3C	
cols=	✔	✔	✔	✔	✔*	✔	2	W3C	

Etichetele și atributele care sunt scrise cu litere **aldine** sunt incluse în limbajul HTML 4.0, iar etichetele între paranteze drepte sunt opționale. Tabelul este aranjat în ordine alfabetică, pentru o regăsire mai ușoară, dar tipurile Structure/Block/Text ale etichetelor sunt identificate prin Ⓢ, Ⓑ, Ⓣ după eticheta propriu-zisă.

2,3,4 etc. = Numărul versiunii HTML condensate; 3+ = versiunea HTML 3.2; ✔ = disponibil pentru toate versiunile curente; ↓ nerecomandat – versiunea HTML; ✘ = depășit; Ⓟ = propriu/neconform standardului; *= implementare parțială sau parțial neconform standardului; NS=Non Specified.

Eticheta	*MSIE*	*Netscp*	*NCSA*	*Opera*	*Lynx*	*Emacs*	*HTML*	*Sponsor*	*Status*
<TEXTAREA>...</TEXTAREA> *continuare*									
datafld=	✔	-	-	-	-	-	Ⓟ	MS	
datasrc=	✔	-	-	-	-	-	Ⓟ	MS	
dir=	-	-	-	-	-	-	4	W3C	
disabled=	4	-	-	-	-	-	3/4	W3C	
error=	-	-	-	-	-	-	3	W3C	
id=	4	3	-	-	✔	✔	3/4	W3C	
lang=	-	✔3/4	-	-	✔	✔	3/4	W3C	
name=	✔	✔	✔	✔	✔	✔	2-4	W3C	
readonly=	4	-	-	-	-	-	4	W3C	
rows=	✔	✔	✔	✔	✔*	✔	2-4	W3C	
style=	4	4	-	-	-	-	4	W3C	
tabindex=	4	4*	-	-	-	-	4	W3C	
title=	4	-	-	-	-	-	4	W3C	
wrap=	-	✔	-	✔	-	-	Ⓟ	NS	✘4
(virtual, physical, none)									
event=	4	-	-	-	-	-	4	W3C	
<TFOOT>[</TFOOT>] Ⓢ	✔3/4	4	-	3	-	-	4	W3C	
align=	4	-	-	-	-	-	4	W3C	
(center, left, right)									
bgcolor=	4	-	-	-	-	-	Ⓟ	MS	
char=	-	-	-	-	-	-	4	W3C	
charoff=	-	-	-	-	-	-	4	W3C	
dir=	-	-	-	-	-	-	4	W3C	
id=	4	-	-	-	-	-	4	W3C	

Eticheta	*MSIE*	*Netscp*	*NCSA*	*Opera*	*Lynx*	*Emacs*	*HTML*	*Sponsor*	*Status*
lang=	-	-	-	-	-	-	4	W3C	
style=	4	-	-	-	-	-	4	W3C	
title=	4	-	-	-	-	-	4	W3C	
valign=	4	-	-	-	-	-	4	W3C	
(baseline, bottom, center, top)									
event=	4	-	-	-	-	-	4	W3C	
<TH></TH> Ⓢ	✔	✔	✔	✔	✔	✔	3+/4	W3C	
align=	✔*	✔	✔*	✔	✔*	✔	3+/4	W3C	
(center, left, right)							3	W3C	✘3+
(decimal)							3	W3C	✘3+
(justify)							3/4	W3C	
axes=	-	-	-	-	-	-	3/4	W3C	
axis=	-	-	-	-	-	-	Ⓟ		✘4
background=	✔	✔	-	-	-	-	4	W3C	
bgcolor=	✔	✔	-	✔	-	-	Ⓟ	MS	
bordercolor=	4	-	-	-	-	-	Ⓟ	MS	
bordercolordark=	4	-	-	-	-	-	Ⓟ	MS	
bordercolorlight=	4	-	-	-	-	-	4	W3C	
char=	-	-	-	-	-	-			

Etichetele și atributele care sunt scrise cu litere **aldine** sunt incluse în limbajul HTML 4.0, iar etichetele între paranteze drepte sunt opționale. Tabelul este aranjat în ordine alfabetică, pentru o regăsire mai ușoară, dar tipurile Structure/Block/Text ale etichetelor sunt identificate prin Ⓢ, Ⓑ, Ⓣ după eticheta propriu-zisă.

2,3,4 etc. = Numărul versiunii HTML condensate; 3+ = versiunea HTML 3.2; ✔ = disponibil pentru toate versiunile curente; ↓ nerecomandat – versiunea HTML; ✘ = depășit; Ⓟ = propriu/neconform standardului; *= implementare parțială sau parțial neconform standardului; NS=Non Specified.

Eticheta	*MSIE*	*Netscp*	*NCSA*	*Opera*	*Lynx*	*Emacs*	*HTML*	*Sponsor*	*Status*
<TH></TH> *continuare*									
charoff=	-	-	-	-	-	-	4	W3C	
class=	-	-	-	-	-	-	4	W3C	
colspan=	✔	✔	✔	✔	-	-	3+/4	W3C	
dir=	-	-	-	-	-	-	4	W3C	
dp=	-	-	-	-	-	-	3	W3C	✘3+
height=	✔	✔	-	✔	-	-	3+	W3C	✘4
id=	4	3	-	-	-	-	3	W3C	
lang=	-	3	-	-	-	-	3	W3C	
nowrap	✔	✔	✔	✔	-	-	3+	W3C	
rowspan=	✔	✔	✔	✔	-	-	3+	W3C	
style=	4	4	-	-	-	-	4	W3C	
title=	4	-	-	-	-	-	4	W3C	
valign=	✔	✔	-	✔	-	-	3+	W3C	
(baseline, bottom, center, top)									
width=	✔	✔	✔	✔	-	-	3+	W3C	✘4
event=	4	-	-	-	-	-	4	W3C	
<THEAD> ...[</THEAD>] Ⓢ	✔3/4	4	-	3	-	-	4	W3C	
align=	4	-	-	-	-	-	4	W3C	
(center, left, right)									
bgcolor=	4	-	-	-	-	-	Ⓟ	MS	
char=	-	-	-	-	-	-	4	W3C	
charoff=	-	-	-	-	-	-	4	W3C	
class=	-	-	-	-	-	-	4	W3C	
dir=	-	-	-	-	-	-	4	W3C	

Eticheta	*MSIE*	*Netscp*	*NCSA*	*Opera*	*Lynx*	*Emacs*	*HTML*	*Sponsor*	*Status*
id=	4	-	-	-	-	-	4	W3C	
lang=	-	-	-	-	-	-	4	W3C	
style=	4	-	-	-	-	-	4	W3C	
title=	4	-	-	-	-	-	4	W3C	
valign=	4	-	-	-	-	-	4	W3C	
(baseline, bottom, center, top)									
event=	4	-	-	-	-	-	4	W3C	
<TITLE></TITLE> Ⓢ	✔	✔	✔	✔	✔	✔	2-4	W3C	

OBSERVAȚIE: Deoarece în limbajul HTML 3.2, pentru fiecare document HTML este necesar un singur container `<TITLE>`, deși majoritatea browserelor nu impun această cerință. Este o idee bună să includeți un titlu sugestiv în fiecare document, dar evitați „trucurile" cum ar fi titlurile multiple pseudoanimate.

dir=	-	-	-	-	-	-	4	W3C	
lang=	-	-	-	-	-	-	4	W3C	
<TR>[</TR>] Ⓢ	✔	✔	✔	✔	✔*	✔	3+/4	W3C	
align=	4*	-	-	-	-	-	3+/4	W3C	
(center, left, right)							3	W3C	✘3+
(justify)							4	W3C	
bgcolor=	4	4	-	3	-	-	Ⓟ	MS	
bordercolor=	4	-	-	-	-	-	Ⓟ	MS	
bordercolordark=	4	-	-	-	-	-			

Etichetele și atributele care sunt scrise cu litere **aldine** sunt incluse în limbajul HTML 4.0, iar etichetele între paranteze drepte sunt opționale. Tabelul este aranjat în ordine alfabetică, pentru o regăsire mai ușoară, dar tipurile Structure/Block/Text ale etichetelor sunt identificate prin Ⓢ, Ⓑ, Ⓣ după eticheta propriu-zisă.

2,3,4 etc. = Numărul versiunii HTML condensate; 3+ = versiunea HTML 3.2; ✔ = disponibil pentru toate versiunile curente; ↓ nerecomandat – versiunea HTML; ✘ = depășit; Ⓟ = propriu/neconform standardului; *= implementare parțială sau parțial neconform standardului; NS=Non Specified.

Eticheta	*MSIE*	*Netscp*	*NCSA*	*Opera*	*Lynx*	*Emacs*	*HTML*	*Sponsor*	*Status*
<TR>[</TR>] *continuare*									
bordercolorlight=	4	-	-	-	-	-	Ⓟ	MS	
char=	-	-	-	-	-	-	4	W3C	
charoff=	-	-	-	-	-	-	4	W3C	
class=	-	✔	-	-	-	-	3/4	W3C	
dir=	-	-	-	-	-	-	4	W3C	
dp=	-	-	-	-	-	-	3	W3C	✘3+
height=	4	-	-	-	-	-	3+	W3C	✘4
id=	4	✔	-	-	-	-	3/4	W3C	
lang=	-	✔	-	-	-	-	3/4	W3C	
nowrap	-	✔	-	✔	-	-	3	W3C	
style=	4	4	-	-	-	-	4	W3C	
title=	4	-	-	-	-	-	4	W3C	
valign= (baseline, bottom, center, top)	✔	✔	✔	✔	✔*	✔	3/4	W3C	
vspace=	✔	✔	✔	✔	-	-	Ⓟ	NS	✘4
event=	4	-	-	-	-	-	4	W3C	
<TT></TT> Ⓣ	✔	✔	✔	✔	✔*	✔	2-4	W3C	
class=	-	✔	-	-	✔	✔	3/4	W3C	
dir=	-	-	-	-	-	-	4	W3C	
id=	3/4	✔	-	-	✔	✔	3/4	W3C	
lang=	-	✔	-	-	✔	✔	3/4	W3C	
style=	4	4	-	-	-	-	4	W3C	
title=	4	-	-	-	-	-	4	W3C	
event=	4	-	-	-	-	-	4	W3C	

Eticheta	*MSIE*	*Netscp*	*NCSA*	*Opera*	*Lynx*	*Emacs*	*HTML*	*Sponsor*	*Status*
<U></U> Ⓣ	✔	✔	✔	✔	✔*	✔	3/4	W3C	↓4
class=	-	✔	-	-	✔	✔	3/4	W3C	
dir=	-	-	-	-	-	-	4	W3C	
id=	✔	✔	-	-	✔	✔	3/4	W3C	
lang=	-	✔	-	-	✔	✔	3/4	W3C	
style=	4	4	-	-	-	-	4	W3C	
title=	4	-	-	-	-	-	4	W3C	
event=	4	-	-	-	-	-	4	W3C	
<UL></UL> Ⓢ	✔	✔	✔	✔	✔	✔	2-4	W3C	
align= (center, left, right)	4	-	-	-	✔	-			✘4
class=	-	3	-	-	✔	✔	3/4	W3C	
clear= (left, right, all)	-	-	-	-	-	-	3	W3C	✘3+
compact	✔*	✔*	✔*	✔*	✔*	✔*	2-4	W3C	↓4
dingbat=	-	-	-	-	✔*	✔	3	W3C	✘3+
dir=	-	-	-	-	-	-	4	W3C	
id=	4	3	-	-	-	-	3/4	W3C	
lang=	-	3	-	-	-	-	3/4	W3C	

Etichetele și atributele care sunt scrise cu litere **aldine** sunt incluse în limbajul HTML 4.0, iar etichetele între paranteze drepte sunt opționale. Tabelul este aranjat în ordine alfabetică, pentru o regăsire mai ușoară, dar tipurile Structure/Block/Text ale etichetelor sunt identificate prin Ⓢ, Ⓑ, Ⓣ după eticheta propriu-zisă.

2,3,4 etc. = Numărul versiunii HTML condensate; 3+ = versiunea HTML 3.2; ✔ = disponibil pentru toate versiunile curente; ↓ nerecomandat – versiunea HTML; ✘ = depășit; Ⓟ = propriu/neconform standardului; *= implementare parțială sau parțial neconform standardului; NS=Non Specified.

Eticheta	*MSIE*	*Netscp*	*NCSA*	*Opera*	*Lynx*	*Emacs*	*HTML*	*Sponsor*	*Status*
<UL></UL> *continuare*									
md=	-	-	-	-	-	-	3	W3C	✗3+
plain	-	-	-	-	-	-	3	W3C	✗3+
src=	4	-	-	-	-	-	3	W3C	✗3+
style=	4	4	-	-	-	-	4	W3C	
title=	4	-	-	-	-	-	4	W3C	
type=	4	-	-	-	✔*	✔	3+/4	W3C	
(disk, square, circle)									
wrap=	-	-	-	-	-	-	3	W3C	✗3+
(vert, horiz)									
event=	4	-	-	-	-	-	4	W3C	
<VAR></VAR> [T]	✔	✔	✔	✔	✔*	✔	3+/4	W3C	
class=	-	✔	-	-	✔	✔	3/4	W3C	
dir=	-	-	-	-	-	-	4	W3C	
id=	✔	✔	-	-	✔	✔	3/4	W3C	
lang=	-	✔	-	-	✔	✔	3/4	W3C	
style=	✔	✔	-	-	-	-	4	W3C	
title=	4	-	-	-	-	-	4	W3C	
event=	4	-	-	-	-	-	4	W3C	
<WBR> [S] [T]	✔	✔	-	✔	-	✔	[P]	NS	
id=	4	3	-	-	-	✔			
style=	4	4	-	-	-	✔			
title=	4	-	-	-	-	✔			

BIBL. CENTR. UNIV. „M. EMINESCU" IAȘI INFORMATICĂ

Eticheta	*MSIE*	*Netscp*	*NCSA*	*Opera*	*Lynx*	*Emacs*	*HTML*	*Sponsor*	*Status*
<XMP></XMP> [S]	✔	✔	✔	✔	✔	✔	2	W3C	↓2✗4
id=	4	3	-	-	-	-	3	W3C	
style=	4	4	-	-	-	-	3	W3C	
title=	4	-	-	-	-	-	3	W3C	
event=	4	-	-	-	-	-	4	W3C	

Evenimente intrinseci

Eveniment intrinsec	*MSIE*	*Netscp*	*NCSA*	*Opera*	*Lynx*	*Emacs*	*HTML*	*Sponsor*	*Status*
onafterupdate	4								
onbeforeupdate	4								
onblur	4	4				4		W3C	
onchange	-	4				4		W3C	
onclick	4	4				4		W3C	
ondblclick	4	4				4		W3C	
onfocus	4	4				4		W3C	
onhelp	4								
onkeydown	4	4				4		W3C	
onkeypress	4	4				4		W3C	
onkeyup	4	4				4		W3C	

Etichetele și atributele care sunt scrise cu litere **aldine** sunt incluse în limbajul HTML 4.0, iar etichetele între paranteze drepte sunt opționale. Tabelul este aranjat în ordine alfabetică, pentru o regăsire mai ușoară, dar tipurile Structure/Block/Text ale etichetelor sunt identificate prin [S], [B], [T] după eticheta propriu-zisă.

2,3,4 etc. = Numărul versiunii HTML condensate; 3+ = versiunea HTML 3.2; ✔ = disponibil pentru toate versiunile curente; ↓ nerecomandat – versiunea HTML; ✗ = depășit; [P] = propriu/neconform standardului; *= implementare parțială sau parțial neconform standardului; NS=Non Specified.

Etichetă ↓	*MSIE*	*Netscp*	*NCSA*	*Opera*	*Lynx*	*Emacs*	*HTML*	*Sponsor*	*Status*
onmousedown	4	4				4		W3C	
onmousemove	4	4				4		W3C	
onmouseout	4	4				4		W3C	
onmouseover	4	4				4		W3C	
onmouseup	4	4				4		W3C	
onsubmit	4	4				4		W3C	
onreset	4	4				4		W3C	

OBSERVAȚIE

Rețineți că toate evenimentele sunt definite pentru fiecare etichetă.

OBSERVAȚIE

Pentru cele mai noi informații oficiale despre HTML 4.0 consultați pagina Web a Consorțiului World Wide Web la adresa `http://www.w3.org/`.

Pentru a afla mai multe despre caracteristicile browserelor, vizitați următoarele situri Web.

Emacs W3:
`http://www.cs.indiana.edu/elisp/w3/docs.html`

Lynx:
`http://www.crl.com/~subir/lynx.html`

Microsoft Internet Explorer:
`http://www.microsoft.com/ie/`

Netscape Navigator:
`http://www.netscape.com/`

NCSA Mosaic:
`http://www.ncsa.uiuc.edu/SDG/Software/Mosaic/`

Opera:
`http://www.operasoftware.com`

Entități HTML 4.0 de tip Escape

Următoarele caractere ar trebui să fie întotdeauna de tip „Escape" în codul scris de dumneavoastră pentru a evita problemele greu de identificat deoarece ele sunt semnificative pentru marcajul din HTML.

Entitatea numită	*Entitatea numerică*	*Descriere*
"	"	" (semnul două ghilimele)
&	&	& (ampersand)
>	>	> (mai mare decât)
<	<	< (mai mic decât)

Deși problema s-ar putea să nu fie afișată ca o eroare de către browserul dumneavoastră, alte browsere ar putea să nu aibă încorporată facilitatea de „revenire din eroare".

Culori stabilite pentru HTML 4.0

Culorile din lista următoare sunt aproximări ale culorilor Windows standard așa cum sunt afișate pe un monitor color cu 16 culori, un fel de cel mai mic numitor. Gama completă de culori ce pot fi afișate pe monitor este disponibilă prin sintaxa perechii hexa, pentru a preciza culoarea stabilită.

Am adăugat numele culorii de proces (de tipărire) adesea folosit pentru a defini culorile cu scădere pure (CYMK), cât și culorile primare aditive (de monitor) din SUA. Totuși, ele nu sunt nume oficiale în standard; de fapt, ele intră parțial în conflict cu standardul. Din cauza multitudinii de nume diferite folosite de producătorii de browsere, probabil este cel mai bine să folosiți valorile numerice neambigue, și nu numele autoexplicative superficiale, așa că nu le-am prezentat în format de tip „escape".

Numele culorii	*Sintaxa perechii hexa*	*Culoarea procesului*
Black (negru)	= #000000	K = (Black)
Navy (bleumarin)	= #000080	
Blue (albastru)	= #0000FF	B = (Blue)
Green (verde)	= #008000	
Teal (albastru-verzui)	= #008080	
Lime (vernil)	= #00FF00	G = (Green)
Aqua (azuriu)	= #00FFFF	C = (Cyan)
Maroon (maron)	= #800000	
Purple (purpuriu)	= #800080	
Olive (oliv)	= #808000	
Gray (gri)	= #808080	
Silver (argintiu)	= #C0C0C0	
Red (roșu)	= #FF0000	
Fuchsia (fucsia)	= #FF00FF	M = (Magenta)
Yellow (galben)	= #FFFF00	Y = (Yellow)
White (alb)	= #FFFFFF	

Entități HTML 4.0 de tip caracter

Tabelul următor oferă aceleași date în două moduri diferite, în ordinea ASCII (majusculele sunt selectate înaintea minusculelor) și în ordinea numerică. Motivul este că, atunci când cunoașteți numele caracterului, puteți găsi echivalentul numeric care se potrivește și invers, când cunoașteți numărul, puteți ști ce reprezintă el.

SUGESTIE

Regula generală în HTML este că nu contează dacă literele sunt majuscule sau minuscule, dar numele caracterelor fac excepție deoarece ele aparțin altui standard – ISO. Puține browsere acceptă toate caracterele, totuși, dacă sunteți atașat de o anumită glifă obscură, probabil este mai sigur să folosiți valoarea numerică și să o testați pe mai multe browsere înainte de a o lansa pe Web.

OBSERVAȚIE

Portions © International Organization for Standardization, 1986.

Este permisă copierea de orice fel pentru folosirea cu sistemele conforme cu SGML și cu aplicații așa cum s-a definit în ISO 8879 cu condiția ca prezenta observație să fie inclusă în toate exemplarele.

Ordinea ASCII

Nume	*Valoare*	*Descriere*
Æ	Æ	Diftongul majuscul AE (ligatură)
Á	Á	Majuscula A, cu accent ascuțit
Â	Â	Majuscula A, cu accent circumflex
À	À	Majuscula A, cu accent grav
Α	Α	Litera majusculă grecească alfa, Unicode: 0391
Å	Å	Majuscula A, cu cerculeț
Ã	Ã	Majuscula A, cu tildă
Ä	Ä	Majuscula A, cu umlaut
Β	Β	Litera majusculă grecească beta, Unicode: 0392
Ç	Ç	Majuscula C, cu sedilă
Χ	Χ	Litera majusculă grecească chi, Unicode: 03A7
‡	‡	Cruce dublă, Unicode: 2021
Δ	Δ	Litera majusculă grecească delta, Unicode: 0394
Ð	Ð	Majuscula Eth, alfabetul islandez
É	É	Majuscula E, cu accent ascuțit
Ê	Ê	Majuscula E, cu accent circumflex
È	È	Majuscula E, cu accent grav
Ε	Ε	Litera majusculă grecească epsilon, Unicode: 0395
Η	Η	Litera majusculă grecească eta, Unicode: 0397
Ë	Ë	Majuscula E, cu umlaut
Γ	Γ	Litera majusculă grecească gama, Unicode: 0393
Í	Í	Majuscula I, cu accent ascuțit
Î	Î	Majuscula I, cu accent circumflex

Nume	*Valoare*	*Descriere*
Ì	Ì	Majuscula I, cu accent grav
Ι	Ι	Litera majusculă grecească iota, Unicode: 0399
Ï	Ï	Majuscula I, cu umlaut
Κ	Κ	Litera majusculă grecească kapa, Unicode: 039A
Λ	Λ	Litera majusculă grecească lambda, Unicode: 039B
Μ	Μ	Litera majusculă grecească miu, Unicode: 039C
Ñ	Ñ	Majuscula N, cu tildă
Ν	Ν	Litera majusculă grecească niu, Unicode: 039D
Œ	Œ	Ligatura latină majusculă OE, Unicode: 0152
Ó	Ó	Majuscula O, cu accent ascuțit
Ô	Ô	Majuscula O, cu accent circumflex
Ò	Ò	Majuscula O, cu accent grav
Ω	Ω	Litera majusculă grecească omega, Unicode: 03A9
Ο	Ο	Litera majusculă grecească omicron, Unicode: 039F
Ø	Ø	Majuscula O, cu bară
Õ	Õ	Majuscula O, cu tildă
Ö	Ö	Majuscula O, cu umlaut
Φ	Φ	Litera majusculă grecească fi, Unicode: 03A6
Π	Π	Litera majusculă grecească pi, Unicode: 03A0
″	″	Apostrof dublu, secunde, inci, Unicode: 2033
Ψ	Ψ	Litera majusculă grecească psi, Unicode: 03A8
Ρ	Ρ	Litera majusculă grecească ro, Unicode: 03A1
Š	Š	Majuscula latină S cu semn diacritic deasupra, Unicode: 0160
Σ	Σ	Litera majusculă grecească sigma, Unicode: 03A3
Þ	Þ	Majuscula THORN, alfabetul islandez
Τ	Τ	Litera majusculă grecească tau, Unicode: 03A4
Θ	Θ	Litera majusculă grecească teta, Unicode: 0398
Ú	Ú	Majuscula U, cu accent ascuțit
Û	Û	Majuscula U, cu accent circumflex
Ù	Ù	Majuscula U, cu accent grav
Υ	Υ	Litera majusculă grecească upsilon, Unicode: 03A5
Ü	Ü	Majuscula U, cu umlaut
Ξ	Ξ	Litera majusculă grecească xi, Unicode: 039E
Ý	Ý	Majuscula Y, cu accent ascuțit
Ÿ	Ÿ	Majuscula latină y, cu umlaut, Unicode: 0178
Ζ	Ζ	Litera majusculă grecească zeta, Unicode: 0396
á	á	Litera minusculă a, cu accent ascuțit
â	â	Litera minusculă a, cu accent circumflex
´	´	Accent ascuțit

(continuare)

Nume	Valoare	Descriere
æ	æ	Diftongul format din minusculele ae (ligatură)
à	à	Litera minusculă a, cu accent grav
ℵ	ℵ	Simbolul alef, primul număr cardinal transfinit, Unicode: 2135
α	α	Litera minusculă grecească alfa, Unicode: 03B1
&	&	Ampersand, Unicode: 0026
∧	⊥	Semnul operației logice „și", Unicode: 2227
∠	∠	Unghi, Unicode: 2220
å	å	Litera minusculă a, cu cerculeț
&asmp;	≈	Semnul aproximativ egal cu, asimptotic la, Unicode: 2248
ã	ã	Litera minusculă a, cu tildă
ä	ä	Litera minusculă a, cu umlaut
„	„	Ghilimele duble de jos, de tip 9, Unicode: 201E
β	β	Litera minusculă grecească beta, Unicode: 03B2
¦	¦	Bară verticală
•	•	Marcaj de listă, cerc mic negru,Unicode: 2022
∩	∩	Intersecție, verzală, Unicode: 2229
ç	ç	Litera minusculă c, cu sedilă
¸	¸	Sedilă
¢	¢	Semnul pentru cent
χ	χ	Litera minusculă grecească chi, Unicode: 03C7
ˆ	ˆ	Accent circumflex, Unicode: 02C6
♣	♣	Treflă neagră, trifoi, Unicode: 266
≅	≅	Aproximativ egal cu, Unicode: 2245
©	©	Semnul de copyright
↵	↵	Săgeată în jos îndoită spre stânga, carriage return, Unicode: 21B5
∪	∪	Reuniune, Unicode: 222A
¤	¤	Semnul general pentru valută
⇓	⇓	Săgeată dublă în jos, Unicode: 21D3
†	†	Cruce, Unicode: 2020
↓	↓	Săgeată în jos, Unicode: 2193
°	°	Semnul de grad
δ	δ	Litera minusculă grecească delta, Unicode: 03B4
♦	♦	Caro negru, Unicode: 2666
÷	÷	Semnul de împărțire
é	é	Litera minusculă e, cu accent ascuțit
ê	ê	Litera minusculă e, cu accent circumflex
&egrave	è	Litera minusculă e, cu accent grav
∅	∅	Mulțime vidă, mulțime nulă, diametru, Unicode: 2205
		Spațiul em, Unicode: 2003
		Spațiul en, Unicode: 2002

Nume	Valoare	Descriere
ε	ε	Litera minusculă grecească epsilon, Unicode: 03B5
≡	≡	Identic cu, Unicode: 2261
η	η	Litera minusculă grecească eta, Unicode: 03B7
ð	ð	Litera minusculă eth, alfabetul islandez
ë	ë	Litera minusculă e, cu umlaut
∃	∄	Semnul matematic pentru „există", Unicode: 2203
ƒ	ƒ	Minuscula latină f rond, funcție, florin, Unicode: 0192
∀	∀	Semnul matematic „pentru orice", Unicode: 2200
½	½	Jumătate
¼	¼	Un sfert
¾	¾	Trei sferturi
⁄	⁄	Bara de fracție, Unicode: 2044
γ	γ	Litera minusculă grecească gama, Unicode: 03B3
≥	≥	Semnul mai mare sau egal cu, Unicode: 2265
>	>	Semnul mai mare decât, Unicode: 003E
⇔	⇔	Săgeată dublă stânga-dreapta, Unicode: 21D4
↔	↔	Săgeată stânga-dreapta, Unicode: 2194
♥	♥	Cupă neagră, Valentine, Unicode:2665
…	…	Puncte de suspensie orizontale, trei puncte, Unicode: 2026
í	í	Litera minusculă i, cu accent ascuțit
î	î	Litera minusculă i, cu accent circumflex
¡	¡	Semn de exclamație inversat
ì	ì	Litera minusculă i, cu accent grav
ℑ	ℑ	Litera majuscula I gotică, semnul pentru partea imaginară, Unicode: 2111
∞	∞	Semnul infinit, Unicode: 221E
∫	∫	Semnul pentru integrală, Unicode: 222B
ι	ι	Litera minusculă grecească iota, Unicode: 03B9
¿	¿	Semnul de întrebare inversat
∈	Ῥ	Semnul de „aparține", Unicode: 2208
ï	ï	Litera minusculă i, cu umlaut
κ	κ	Litera minusculă grecească kapa, Unicode: 03BA
⇐	⇐	Săgeată dublă la stânga, Unicode: 21D0
λ	λ	Litera minusculă grecească lambda, Unicode: 03BB
⟨	〈	Paranteză unghiulară la stânga, Unicode: 2329
«	«	Ghilimele unghiulare, stânga
←	←	Săgeată spre stânga, Unicode: 2190
⌈	⌈	Liniuță sus cu margine în stânga, Unicode: 2308
“	“	Ghilimele duble la stânga, Unicode: 201C
≤	≤	Mai mic sau egal cu, Unicode: 2264

(continuare)

Nume	Valoare	Descriere
⌊	⌊	Liniuță jos cu margine în stânga, Unicode: 230A
∗	∗	Asterisc, Unicode: 2217
◊	◊	Romb, Unicode: 25CA
‎	‎	Săgeată stânga-dreapta, Unicode: 200E RFC 2070
‹	‹	Apostrof unghiular la stânga, Unicode: 2039
‘	‘	Apostrof la stânga, Unicode: 2018
<	<	Semnul mai mic decât, Unicode: 003C
¯	¯	Semn pentru vocală lungă
—	—	Liniuță em, Unicode: 2014
µ	µ	Semnul micro
·	·	Punct de mijloc
−	−	Semnul minus, Unicode: 2212
μ	μ	Litera minusculă grecească miu, Unicode: 03BC
∇	∇	Semnul nabla, Unicode: 2207
		Spațiu neîntrerupt
–	–	Liniuță en, Unicode: 2013
≠	≠	Semnul diferit de, Unicode: 2260
∋	∋	Semnul „conține ca membru", Unicode: 220B
¬	¬	Fără semn
∉	∉	Semnul „nu aparține", Unicode: 2209
⊄	⊄	Semnul nu este inclus, Unicode: 2284
ñ	ñ	Litera minusculă n, cu tildă
ν	ν	Litera minusculă grecească niu, Unicode: 03BD
ó	ó	Litera minusculă o, cu accent ascuțit
ô	ô	Litera minusculă o, cu accent circumflex
œ	œ	Ligatura din litere minuscule latine oe, Unicode: 0153
ò	ò	Litera minusculă o, cu accent grav
‾	‾	Liniuță sus, spațiere în partea de sus, Unicode: 203E
ω	ω	Litera minusculă grecească omega, Unicode: 03C9
ο	ο	Litera minusculă grecească omicron, Unicode: 03BF
⊕	⊕	Semnul plus într-un cerc, sumă directă, Unicode: 2295
∨	⊦	Semnul SAU logic, Unicode: 2228
ª	ª	Indicator ordinal, feminin
º	º	Indicator ordinal, masculin
ø	ø	Lintera minusculă o, barată
õ	õ	Litera minusculă o, cu tildă
⊗	⊗	Semnul de înmulțire încercuit, produs vectorial, Unicode: 2297
ö	ö	Litera minusculă o, cu umlaut
¶	¶	Simbolul de paragraf

Nume	Valoare	Descriere
∂	∂	Diferențială, Unicode: 2202
‰	‰	Semnul „la mie", Unicode: 2030
⊥	⊥	Ortogonal, perpendicular, Unicode: 22A5
φ	φ	Litera minusculă grecească fi, Unicode: 03C6
π	π	Litera minusculă grecească pi, Unicode: 03C0
ϖ	ϖ	Simbolul grecesc pi, Unicode: 03D6
±	±	Semnul plus - minus
£	£	Semnul de liră sterlină
′	′	Apostrof, minute, picioare, Unicode: 2032
∏	∏	Produs după n, semnul de produs, Unicode: 220F
&rop;	∝	Semnul proporțional cu, Unicode: 221D
ψ	ψ	Litera minusculă grecească psi, Unicode: 03C8
"	"	Ghilimele, Unicode: 0022
⇒	⇒	Săgeată dublă spre dreapta, Unicode: 21D2
√	√	Rădăcină pătrată, semnul radical, Unicode: 221A
⟩	〉	Paranteză unghiulară la dreapta, Unicode: 232A
»	»	Ghilimele unghiulare, dreapta
→	→	Săgeată spre dreapta, Unicode: 2192
⌉	⌉	Liniuță sus cu margine în dreapta, Unicode: 2309
”	”	Ghilimele în dreapta, Unicode: 201D
ℜ	ℜ	Litera majusculă gotică R , simbolul pentru mulțimea reală, Unicode: 211C
®	®	Semnul de marcă înregistrată
⌋	⌋	Liniuță jos cu margine în dreapta, Unicode: 230B
ρ	ρ	Litera minusculă grecească ro, Unicode: 03C1
‏	‏	Semnul stânga-dreapta, Unicode: 200F RFC 2070
›	›	Apostrof unghiular la dreapta, Unicode: 203A
’	’	Apostrof la dreapta, Unicode: 2019
‚	‚	Apostrof de jos, de tip 9, Unicode: 201A
š	š	Litera minusculă latină s cu semn diacritic deasupra, Unicode: 0161
⋅	⋅	Operatorul punct, Unicode: 22C5
§	§	Semnul de secțiune
		Semnul de despărțire în silabe soft
σ	σ	Litera minusculă grecească sigma, Unicode: 03C3
ς	ς	Litera minusculă grecească sigma final, Unicode: 03C2
∼	∼	Operator tildă, variază cu, similar cu, Unicode: 223C
♠	♠	Pică neagră, Unicode: 2660
⊂	⊂	Semnul „inclus în", Unicode: 2282
⊆	⊆	Semnul „inclus în" sau egal cu, Unicode: 2286
∑	∑	Sumă după n, Unicode: 2211

(continuare)

Nume	*Valoare*	*Descriere*
⊃	⊃	Semnul „include", Unicode: 2283
¹	¹	Exponent unu
²	²	Exponent doi
³	³	Exponent trei
⊇	⊇	Semnul „include" sau egal cu, Unicode: 2287
ß	ß	Litera minusculă s ascuțit, sz din alfabetul german (ligatură)
τ	τ	Litera minusculă grecească tau, Unicode: 03C4
∴	∴	Semnul „rezultă", Unicode: 2234
θ	θ	Litera minusculă grecească teta, Unicode: 03B8
ϑ	ϑ	Litera minusculă grecească simbol teta, Unicode: 03D1
		Spațiu subțire, Unicode: 2009
þ	þ	Semnul thorn scris cu minuscule, din alfabetul islandez
˜	˜	Tildă mică, Unicode: 02DC
×	×	Semnul de înmulțire
™	™	Semnul marca fabricii (™), Unicode: 2122
⇑	⇑	Săgeată dublă în sus, Unicode: 21D1
ú	ú	Litera minusculă u, cu accent ascuțit
↑	↑	Săgeată în sus, Unicode: 2191
û	û	Litera minusculă u, cu accent circumflex
ù	ù	Litera minusculă u, cu accent grav
¨	¨	Semnul umlaut (dieresis)
ϒ	ϒ	Simbolul grecesc upsilon rond, Unicode: 03D2
υ	υ	Litera minusculă grecească upsilon, Unicode: 03C5
ü	ü	Litera minusculă u, cu umlaut
℘	℘	Litera majusculă P de mână, semnul pentru definirea puterii, Weierstrass p, Unicode: 2118
ξ	ξ	Litera minusculă grecească xi, Unicode: 03BE
ý	ý	Litera minusculă y, cu accent ascuțit
¥	¥	Semnul yen
ÿ	ÿ	Litera minusculă y, cu umlaut
ζ	ζ	Litera minusculă grecească zeta, Unicode: 03B6
‍	‍	Simbol de atașare, cu lățime zero, Unicode: 200D
‌	‌	Simbol de separare, cu lățime zero, Unicode: 200C

Ordine numerică

Nume	*Valoare*	*Descriere*
"	"	Ghilimele, Unicode: 0022
&	&	Ampersand, Unicode: 0026
<	<	Semnul mai mic decât, Unicode: 003C
>	>	Semnul mai mare decât, Unicode: 003E

Nume	*Valoare*	*Descriere*
		Spațiu neîntrerupt
¡	¡	Semnul de exclamație inversat
¢	¢	Semnul cent
£	£	Semnul liră sterlină
¤	¤	Semnul general de valută
¥	¥	Semnul yen
¦	¦	Bară verticală
§	§	Semnul de secțiune
¨	¨	Umlaut (dieresis)
©	©	Semnul copyright
ª	ª	Indicator ordinal, feminin
«	«	Ghilimele unghiulare, stânga
¬	¬	Fără semn
		Semnul de despărțire în silabe soft
®	®	Semnul de marcă înregistrată
¯	¯	Semnul pentru vocală lungă
°	°	Semnul de grad
±	±	Semnul plus - minus
²	²	Exponent doi
³	³	Exponent trei
´	´	Accentul ascuțit
µ	µ	Semnul micro
¶	¶	Simbolul de paragraf
·	·	Punct de mijloc
¸	¸	Sedilă
¹	¹	Exponent unu
º	º	Indicator ordinal, masculin
»	»	Ghilimele unghiulare dreapta
¼	¼	Semnul un sfert
½	½	Semnul o jumătate
¾	¾	Semnul trei sferturi
¿	¿	Semnul de întrebare inversat
À	&#192	Litera majusculă A, cu accent grav
Á	Á	Litera majusculă A, cu accent ascuțit
Â	Â	Litera majusculă A, cu accent circumflex
Ã	Ã	Litera majusculă A, cu tildă
Ä	Ä	Litera majusculă A, cu umlaut
Å	Å	Litera majusculă A, cu cerculeț
Æ	Æ	Diftongul scris cu majuscule, AE (ligatură)

(continuare)

Nume	Valoare	Descriere
Ç	Ç	Litera majusculă C, cu sedilă
È	È	Litera majusculă E, cu accent grav
É	É	Litera majusculă E, cu accent ascuțit
Ê	Ê	Litera majusculă E, cu accent circumflex
Ë	Ë	Litera majusculă E, cu umlaut
Ì	Ì	Litera majusculă I, cu accent grav
Í	Í	Litera majusculă I, cu accent ascuțit
Î	Î	Litera majusculă I, cu accent circumflex
Ï	Ï	Litera majusculă I, cu umlaut
Ð	Ð	Litera majusculă Eth, din alfabetul islandez
Ñ	Ñ	Litera majusculă N, cu tildă
Ò	Ò	Litera majusculă O, cu accent grav
Ó	Ó	Litera majusculă O, cu accent ascuțit
Ô	Ô	Litera majusculă O, cu accent circumflex
Õ	Õ	Litera majusculă O, cu tildă
Ö	Ö	Litera majusculă O, cu umlaut
×	×	Semnul de înmulțire
Ø	Ø	Litera majusculă O, barat
Ù	Ù	Litera majusculă U, cu accent grav
Ú	Ú	Litera majusculă U, cu accent ascuțit
Û	Û	Litera majusculă U, cu accent circumflex
Ü	Ü	Litera majusculă U, cu umlaut
Ý	Ý	Litera majusculă Y, cu accent ascuțit
Þ	Þ	Litera majusculă THORN din alfabetul isladez
ß	ß	Litera minusculă s acuțit din alfabetul german (ligatura sz)
à	à	Litera minusculă a, cu accent grav
á	á	Litera minusculă a, cu accent ascuțit
â	â	Litera minusculă a, cu accent circumflex
ã	ã	Litera minusculă a, cu tildă
ä	ä	Litera minusculă a, cu umlaut
å	å	Litera minusculă a, cu cerculeț
æ	æ	Diftongul format din minusculele ae (ligatură)
ç	ç	Litera minusculă c, cu sedilă
è	è	Litera minusculă e, cu accent grav
é	é	Litera minusculă e, cu accent ascuțit
ê	ê	Litera minusculă e, cu accent circumflex
ë	ë	Litera minusculă e, cu umlaut
ì	ì	Litera minusculă i, cu accent grav
í	í	Litera minusculă i, cu accent ascuțit
î	î	Litera minusculă i, cu accent circumflex

Nume	Valoare	Descriere
ï	ï	Litera minusculă i, cu umlaut
ð	ð	Litera minusculă eth, din alfabetul islandez
ñ	ñ	Litera minusculă n, cu tildă
ò	ò	Litera minusculă o, cu accent grav
ó	ó	Litera minusculă o, cu accent ascuțit
ô	ô	Litera minusculă o, cu accent circumflex
õ	õ	Litera minusculă o, cu tildă
ö	ö	Litera minusculă o, cu umlaut
÷	÷	Semnul de împărțire
ø	ø	Litera minusculă o, barat
ù	ù	Litera minusculă u, cu accent grav
ú	ú	Litera minusculă u, cu accent ascuțit
û	û	Litera minusculă u, cu accent circumflex
ü	ü	Litera minusculă u, cu umlaut
ý	ý	Litera minusculă y, cu accent ascuțit
þ	þ	Litera minusculă thorn, din alfabetul islandez
ÿ	ÿ	Litera minusculă y, cu umlaut
Œ	Œ	Ligatura din majuscule latine oe, Unicode: 0152
œ	œ	Ligatura din minuscule latine oe, Unicode: 0153
Š	Š	Litera majusculă latină s cu semn diacritic deasupra, Unicode: 0160
š	š	Litera minusculă latină s cu semn diacritic deasupra, Unicode: 0161
Ÿ	Ÿ	Litera majuscula latină y cu umlaut, Unicode: 0178
ƒ	ƒ	Litera minusculă latină f rond, funcție, florin, Unicode: 0192
ˆ	ˆ	Accent circumflex, Unicode: 02C6
˜	˜	Tildă minusculă, Unicode: 02DC
Α	Α	Litera majusculă grecească alfa, Unicode: 0391
Β	Β	Litera majusculă grecească beta, Unicode: 0392
Γ	Γ	Litera majusculă grecească gama, Unicode: 0393
Δ	Δ	Litera majusculă grecească delta, Unicode: 0394
Ε	Ε	Litera majusculă grecească epsilon, Unicode: 0395
Ζ	Ζ	Litera majusculă grecească zeta, Unicode: 0396
Η	Η	Litera majusculă grecească eta, Unicode: 0397
Θ	Θ	Litera majusculă grecească teta, Unicode: 0398
Ι	Ι	Litera majusculă grecească iota, Unicode: 0399
Κ	Κ	Litera majusculă grecească kapa, Unicode: 039A
Λ	Λ	Litera majusculă grecească lambda, Unicode: 039B
Μ	Μ	Litera majusculă grecească miu, Unicode: 039C
Ν	Ν	Litera majusculă grecească niu, Unicode: 039D
Ξ	Ξ	Litera majusculă grecească xi, Unicode: 039E

(continuare)

Nume	Valoare	Descriere
Ο	Ο	Litera majusculă grecească omicron, Unicode: 039F
Π	Π	Litera majusculă grecească pi, Unicode: 03A0
Ρ	Ρ	Litera majusculă grecească ro, Unicode: 03A1
Σ	Σ	Litera majusculă grecească sigma, Unicode: 03A3
Τ	Τ	Litera majusculă grecească tau, Unicode: 03A4
Υ	Υ	Litera majusculă grecească upsilon, Unicode: 03A5
Φ	Φ	Litera majusculă grecească fi, Unicode: 03A6
Χ	Χ	Litera majusculă grecească chi, Unicode: 03A7
Ψ	Ψ	Litera majusculă grecească psi, Unicode: 03A8
Ω	Ω	Litera majusculă grecească omega, Unicode: 03A9
α	α	Litera minusculă grecească alfa, Unicode: 03B1
β	β	Litera minusculă grecească beta, Unicode: 03B2
γ	γ	Litera minusculă grecească gama, Unicode: 03B3
δ	δ	Litera minusculă grecească delta, Unicode: 03B4
ε	ε	Litera minusculă grecească epsilon, Unicode: 03B5
ζ	ζ	Litera minusculă grecească zeta, Unicode: 03B6
η	η	Litera minusculă grecească eta, Unicode: 03B7
θ	θ	Litera minusculă grecească teta, Unicode: 03B8
ι	ι	Litera minusculă grecească iota, Unicode: 03B9
κ	κ	Litera minusculă grecească kapa, Unicode: 03BA
λ	λ	Litera minusculă grecească lambda, Unicode: 03BB
μ	μ	Litera minusculă grecească miu, Unicode: 03BC
ν	ν	Litera minusculă grecească niu, Unicode: 03BD
ξ	ξ	Litera minusculă grecească xi, Unicode: 03BE
ο	ο	Litera minusculă grecească omicron, Unicode: 03BF
π	π	Litera minusculă grecească pi, Unicode: 03C0
ρ	ρ	Litera minusculă grecească ro, Unicode: 03C1
ς	ς	Litera minusculă grecească sigma final, Unicode: 03C2
σ	σ	Litera minusculă grecească sigma, Unicode: 03C3
τ	τ	Litera minusculă grecească tau, Unicode: 03C4
υ	υ	Litera minusculă grecească upsilon, Unicode: 03C5
φ	φ	Litera minusculă grecească fi, Unicode: 03C6
χ	χ	Litera minusculă grecească chi, Unicode: 03C7
ψ	ψ	Litera minusculă grecească psi, Unicode: 03C8
ω	ω	Litera minusculă grecească omega, Unicode: 03C9
ϑ	ϑ	Litera minusculă grecească simbol teta, Unicode: 03D1
ϒ	ϒ	Litera grecească upsilon rond, Unicode: 03D2
ϖ	ϖ	Simbolul grecesc pi, Unicode: 03D6
		Spațiu en, Unicode: 2002
		Spațiu em, Unicode: 2003

Nume	Valoare	Descriere
		Spațiu subțire, Unicode: 2009
‌	‌	Simbol de separare, cu lățime zero, Unicode: 200C
‍	‍	Simbol de atașare, cu lățime zero, Unicode: 200D
‎	‎	Semnul stânga-dreapta, Unicode: 200E RFC 2070
‏	‏	Semn dreapta-stânga, Unicode: 200F RFC 2070
–	–	Liniuță en, Unicode: 2013
—	—	Liniuță em, Unicode: 2014
‘	‘	Apostrof la stânga, Unicode: 2018
’	’	Apostrof la dreapta, Unicode: 2019
‚	‚	Apostrof de jos, de tip 9, Unicode: 201A
“	“	Ghilimele la stânga, Unicode: 201C
”	”	Ghilimele la dreapta, Unicode: 201D
„	„	Ghilimele duble de jos, de tip 9, Unicode: 201E
†	†	Cruce, Unicode: 2020
‡	‡	Cruce dublă, Unicode: 2021
•	•	Marcaj de listă, cerc mic negru, Unicode: 2022
…	…	Puncte de suspensie orizontale, trei puncte, Unicode: 2026
‰	‰	Semnul „la mie", Unicode: 2030
′	′	Apostrof, minute, picioare, Unicode: 2032
″	″	Ghilimele, secunde, inci, Unicode: 2033
‹	‹	Apostrof unghiular la stânga, Unicode: 2039
›	›	Apostrof unghiular la dreapta Unicode: 203A
‾	‾	Liniuță sus, spațiere în partea de sus, Unicode: 203E
⁄	⁄	Bara de fracție, Unicode: 2044
ℑ	ℑ	Majuscula I gotică, semnul pentru partea imaginară, Unicode: 2111
℘	℘	Litera P de mână, semnul pentru definirea puterii, Weierstrass p, Unicode: 2118
ℜ	ℜ	Majuscula R gotică, simbolul pentru mulțimea reală, Unicode: 211C
™	™	Semnul de marca fabricii (TM), Unicode: 2122
ℵ	ℵ	Simbolul alef (alfabetul ebraic), primul cardinal transfinit, Unicode: 2135
←	←	Săgeata spre stânga, Unicode: 2190
↑	↑	Săgeata în sus, Unicode: 2191
→	→	Săgeata spre dreapta, Unicode: 2192
↓	↓	Săgeata în jos, Unicode: 2193
↔	↔	Săgeata stânga – dreapta, Unicode: 2194
↵	↵	Săgeata în jos îndoită spre stânga, carriage return, Unicode: 21B5
⇐	⇐	Săgeata dublă spre stânga, Unicode: 21D0
⇑	⇑	Săgeată dublă în sus, Unicode: 21D1
⇒	⇒	Săgeata dublă spre dreapta, Unicode: 21D2
⇓	⇓	Săgeată dublă în jos, Unicode: 21D3

(continuare)

Nume	Valoare	Descriere
⇔	⇔	Săgeată dublă stânga-dreapta, Unicode: 21D4
∀	∀	Semnul matematic „pentru orice", Unicode: 2200
∂	∂	Diferențială, Unicode: 2202
∃	∃	Semnul „există", Unicode: 2203
∅	∅	Semnul pentru mulțime vidă, mulțime nulă, diametru, Unicode: 2205
∇	∇	Semnul nabla, Unicode: 2207
∈	∈	Semnul de „aparține", Unicode: 2208
∉	∉	Semnul de „nu aparține", Unicode: 2209
∋	∋	Semnul „conține ca membru", Unicode: 220B
∏	∏	Produs după n, semnul de înmulțire, Unicode: 220F
∑	∑	Adunare după n, Unicode: 2211
−	−	Semnul minus, Unicode: 2212
∗	∗	Operatorul asterisc, Unicode: 2217
√	√	Rădăcină pătrată, semnul radical, Unicode: 221A
∝	∝	Proporțional cu, Unicode: 221D
∞	∞	Semnul pentru infinit, Unicode: 221E
∠	∠	Unghi, Unicode: 2220
∩	∩	Semnul de intersecție, verzală, Unicode: 2229
∪	∪	Semnul de reuniune, simbolul U al reuniunii, Unicode: 222A
∫	∫	Semnul pentru integrală, Unicode: 222B
∴	∴	Semnul „rezultă", Unicode: 2234
∼	∼	Operatorul tildă, variază cu, similar cu, Unicode: 223C
≅	≅	Semnul aproximativ egal cu, Unicode: 2245
≈	≈	Semnul aproape egal cu, asimptotic la, Unicode: 2248
≠	≠	Semnul diferit de, Unicode: 2260
≡	≡	Semnul identic cu, Unicode: 2261
≤	≤	Semnul mai mic sau egal cu, Unicode: 2264
≥	≥	Semnul mai mare sau egal cu, Unicode: 2265
⊂	⊂	Semnul de „inclus în", Unicode: 2282
⊃	⊃	Semnul „include", Unicode, 2283
⊄	⊄	Semnul de „neinclus în", Unicode: 2284
⊆	⊆	Semnul pentru inclus sau egal cu, Unicode: 2286
⊇	⊇	Semnul pentru „include" sau egal cu, Unicode: 2287
⊕	⊕	Semnul plus încercuit, sumă directă, Unicode: 2295
⊗	⊗	Semnul produs, produsul vectorial, Unicode: 2297
∧	⊥	Semnul „ȘI" logic, Unicode: 2227
⊥	⊥	Ortogonal la, perpendicular, Unicode: 22A5
∨	⊦	Semnul „SAU" logic, Unicode: 2228

Nume	Valoare	Descriere
⋅	⋅	Operatorul punct, Unicode: 22C5
⌈	⌈	Liniuță sus cu margine în stânga, Unicode: 2308
⌉	⌉	Liniuță sus cu margine în dreapta, Unicode: 2309
⌊	⌊	Liniuță jos cu margine în stânga, Unicode: 230A
⌋	⌋	Liniuță jos cu margine în dreapta, Unicode: 230B
⟨	〈	Paranteză unghiulară la stânga, Unicode: 2329
⟩	〉	Paranteză unghiulară la dreapta, Unicode: 232A
◊	◊	Romb, Unicode: 25CA
♠	♠	Pică neagră, Unicode: 2660
♣	♣	Treflă neagră, trifoi, Unicode: 2663
♥	♥	Cupă neagră, Valentine, Unicode: 2665
♦	♦	Caro negru, Unicode: 2666

Observații privind tabelele

Deși această anexă era actualizată în timp ce se scria cartea, limbajul HTML 4.0 totuși era în stadiul de proiect, iar caracteristicile browserelor sunt în continuă schimbare. În plus, specificațiile publicate de producătorii de browsere sunt uneori mai mult pline de promisiuni decât exacte, așa că a pretinde că suportul este complet pentru un anumit standard este de obicei mai puțin adevărat. Așa cum am precizat anterior, nici un producător nu aderă la oricare dintre versiunile DTD așa cum se spune.

Pe cât posibil, am verificat documentația existentă a producătorilor, prin testarea codului, dar nu există garanții că o versiune puțin modificată nu va dezvălui mici diferențe referitor la acceptarea atributelor sau etichetelor, nici testul meu nu a fost destul de cuprinzător pentru a descoperi toate comportările neașteptate ce ar putea apărea.

Sunt listate doar cele mai noi versiuni ale fiecărui browser, pentru a fi mai clari și pentru a economisi spațiu. Dacă cele mai noi versiuni de browsere disponibile gratis conțin o anumită caracteristică, se folosește un marcator pentru a indica suportul, în loc de a umple tabelul cu numere de versiuni prea greu de citit. În plus, referințele la standarde HTML încep de la 2.0, deoarece conformitatea cu standardul HTML 1.0 nu mai este recomandată.

Intrările pentru Lynx și Emacs W3 necesită unele explicații. Pentru majoritatea secțiunilor, ambele browsere au motoare interne de analiză care recunosc aproape totul într-un set atotcuprinzător de atribute și etichete HTML pe fiecare platformă. Faptul că producătorul sau modificatorul oricărei versiuni a acestor unelte puternice a ales să facă acea recnoaștere vizibilă pe pagină reprezintă o altă chestiune. Nici Emacs, nici Lynx nu au o comisie centrală de administrare a configurărilor pentru a controla ce intră într-o anumită versiune, iar „caracteristicile lipsă" pot fi ușor adăugate de oricine. De fapt, acesta este modul în care evoluează programele. Ceea ce există în organizație, este consensul, o comunitate mondială de programatori care își dedică timpul și energia pentru a crea ceva ce servește unui scop comun. Aceasta este o lume nouă curajoasă care are astfel de oameni în ea.

Anexa C
Scurtă prezentare a standardului CSS 1

de Rick Darnell

În această anexă

- **Sintaxa de bază 816**
- **Atributele de stil 817**

În această anexă sunt prezentate atributele pe care le puteți folosi pentru a controla aspectul paginilor HTML folosind foile de stil. Consorțiul World Wide Web (W3C) a dat numele „Cascading Style Sheets Level1" (CSS1) standardului actual pentru foile de stil. Recomandările complete de la W3C pentru CSS se găsesc pe Internet la adresa `http://www.w3.org/pub/WWW/TR/REC-CSS1`.

Sintaxa de bază

Când se definește o foaie de stil, toate stilurile folosite respectă aceeași sintaxă de bază. Veți remarca că există numeroase ocazii de a adăuga alte atribute sau membri ai unui grup.

```
SELECTOR [.clasa]  [,SELECTOR2 [.clasa2] ]...
{ atribut1: valoare1   [;
  atribut2: valoare2]  [;
 ... ]                 [;
  atributn: valoaren]  }
```

SELECTOR precizează modul în care este integrat stilul în pagina HTML. Pentru a crea substituri suplimentare, folosește una dintre etichetele existente în HTML, cum ar fi `<CODE>` sau `<P>`, împreună cu o *clasă* opțională. O *clasă* este o submulțime a unui selector și permite ca același element să aibă o diversitate de stiluri. De exemplu, puteți atribui coduri de culoare blocurilor de text pentru a identifica sursele sau vorbitorii.

În plus, pe lângă etichetele din HTML standard, puteți folosi încă două valori pentru un selector, `first line` (prima linie) și `first letter` (prima literă). Valoarea `first line` stabilește stilul pentru prima linie de text sau pentru câteva pasaje dintr-un document, cum ar fi un paragraf sau un bloc de text. Valoarea `first letter` creează litere de început de paragraf și alte efecte speciale pentru prima literă dintr-un document sau pasaj.

Grupurile de selectori și clasele lor sunt separate prin virgule. Oricare membru al grupului va fi definit cu același stil ca și ceilalți membri ai grupului. De exemplu, dacă doriți ca toate titlurile să fie afișate în culoarea roșie, puteți lista atributele de la H1 până la H6 pentru stabilirea culorii în roșu. Celelalte atribute ale etichetelor, cum ar fi mărimea, vor rămâne neschimbate.

O altă opțiune sunt selectorii de context, care indică browserului ce să facă cu o anumită etichetă atunci când o găsește imbricată în eticheta părinte, așa cum este prezentat mai jos:

```
OUTER_SELECTOR INNER_SELECTOR {atribut:valoare}
```

Aceasta înseamnă că, atunci când se folosește `INNER_SELECTOR` (selectorul interior) în interiorul lui `OUTER_SELECTOR` (selectorul exterior), stilul se aplică. Altfel, celelalte apariții ale selectorului interior sunt tratate în conformitate cu configurările prestabilite ale browserului.

După ce ați dat toate definițiile pentru selectori și grupuri, folosiți acoladele pentru a încadra atributele și valorile lor. Puteți asocia fiecare atribut cu valoarea lui, folosind semnul „două puncte" și puteți separa fiecare pereche folosind semnul „punct și virgulă". Valoarea dintr-o definiție, cum ar fi numele tipului de caractere sau o culoare, nu este influențată de majuscule și minuscule. De exemplu, pentru familia de font `font-family`, puteți scrie Garamond, garamond sau GARAMOND și ele vor fi acceptate la fel de către browser.

Ca în cazul tuturor sintaxelor bune, puteți plasa definiții de stil într-un document în trei moduri: cu foaia de stil înglobată, cu o foaie de stil legată și cu foaia de stil inline.

Foaia de stil înglobată

Etichetele `<STYLE>` conțin o foaie de stil înglobată. Din punct de vedere structural, formatul unei pagini HTML care folosește o foaie de stil înglobată este următorul:

```
<HTML>
<HEAD>...</HEAD>
<STYLE>...</STYLE>
<BODY>...</BODY>
</HTML>
```

Etichetele `<STYLE>` conțin lista selectorilor și a stilurilor.

Foaia de stil legată

Foaia de stil legată este de fapt un fișier.css, în care nu se găsește altceva decât un set de etichete `<STYLE>` și conținutul lor. Identificarea fișierului de stiluri într-un document HTML se face folosind eticheta `<LINK>` în titlu, așa cum este prezentat mai jos:

```
<HEAD>
<LINK rel=stylesheet href= "filename.css" type="text/css">
</HEAD>
```

La rulare, browserul încarcă stilul în fișierul `.css` și îl folosește pentru a formata documentul. Dacă pagina HTML conține și o foaie de stil înglobată, iar aceasta intră în conflict cu foaia de stil legată, atunci foaia de stil înglobată are întâietate.

Foaia de stil inline

Ultima variantă, foaia de stil inline, folosește sintaxa foii de stil, deși, din punct de vedere tehnic, nu este o implementare a foii de stil. Această variantă folosește denumirea de foaie de stil pentru a personaliza aparițiile singulare ale etichetelor în document:

```
<TAG style="attribut1:valoare1;...">
```

În principal, aceasta este o modalitate de a personaliza etichetele din HTML după criteriul „caz cu caz". Când folosiți toate cele trei forme ale sintaxei, ele apar în cascadă. Cea mai înaltă prioritate o are foaia de stil inline, urmată de foaia de stil înglobată și apoi de cea legată.

Atributele de stil

La definirea unui selector se folosesc câteva clase de atribute. Următoarea secțiune prezintă fiecare atribut dintr-o anumită clasă.

Atunci când apar conflicte între stilurile dintr-o pagină, regulile se aplică dupa o metodă stabilită. Pentru a determina aspectul unui element în pagină, trebuie să respectați, în ordine, următoarele reguli.

1. Mai întâi, identificați toate declarațiile ce se aplică elementului în cauză. Declarațiile de stil se aplică dacă selectorul se potrivește cu elementul în cauză. Dacă nu se face nici o

declarație, stilul elementului este moștenit de la părintele său. Toate valorile specificate sunt tratate conform cu valorile prestabilite ale browserului.

2. Când există mai multe declarații, orice stil marcat cu `!important` este mai important decât declarațiile nemarcate.

3. Următoarea este prioritatea în funcție de origine. Foile de stil ale autorului au prioritate față de foile de stil ale cititorului, care la rândul lor, au prioritate față de valorile prestabilite ale browserului. Foile de stil legate sunt considerate ca având aceeași origine ca și autorul.

4. Următoarea regulă este specificitatea. Un selector mai specific are prioritate înaintea altor selectori mai puțin specifici. De exemplu, dacă stabiliți corpul BODY cu text negru pe fundal alb și apoi stabiliți BLOCKQUOTE cu text roșu pe fundal alb, atunci un bloc de text din corpul paginii va fi cu roșu, deoarece BLOCKQUOTE este mai specific decât BODY.

5. Sortarea în ordinea în care sunt definite stilurile. Dacă două reguli au aceeași importanță, atunci ultima definiție primează, așa că o foaie de stil legată, aflată la începutul documentului va determina primul stil. Dacă apar diferențe între foaia de stil legată și foaia de stil înglobată, atunci implicit se aplică foaia de stil înglobată. Dacă din anumite motive, foaia de stil înglobată conține două definiții pentru același element, cum ar fi de exemplu două seturi de proprietăți pentru BLOCKQUOTE, atunci ultimul set de proprietăți va avea întâietate.

Folosind aceste reguli, puteți ști precis cum va fi aplicată foaia de stil atunci când va fi interpretată de un browser.

Fonturi

Nu există standarde uzuale pentru tipurile caracterelor și folosirea lor pe diverse calculatoare, așa că va trebui să faceți o alegere atentă și să includeți câteva opțiuni pentru a obține efectul pe care doriți să-l aibă asupra utilizatorului.

Atributul `font-family`

Acest atribut listează familiile fonturilor în ordinea preferințelor, separate prin virgule. Se folosesc două tipuri de variabile: numele familiei și familia generică. Iată un exemplu despre cum se folosește atributul `font-family`.

```
BODY {font-family : Garamond, Palatino, Serif}
```

Un nume de familie reprezintă numele unui tip de caractere, cum ar fi Helvetica, Garamond, Palatino sau Optima. Numele fonturilor se scriu între ghilimele, cu spațiu între ele, cum ar fi de exemplu `"Gil Sans"`. Familia generică reprezintă una dintre cele cinci opțiuni în care sunt clasificate tipurile de caractere, în funcție de stil, și este recomandată ca fiind ultima opțiune într-o listă de familii de fonturi:

- Serif: Fonturi cu accentuări ale capatelor de linie (de exemplu, Times New Roman)
- Sans Serif: Fonturi fără accentuări ale capetelor de linie (de exemplu, Helvetica)
- Cursive: Fonturi care imită caligrafia scrisului de mână (de exemplu, Zapf Chancery)
- Fancy: Fonturi cu utilizare decorativă (de exemplu, ITC Jambalaya sau Bazooka)
- Monospațiate: Fonturi care mențin o spațiere uniformă indiferent de lățimea literei (de exemplu, Courier)

Atributul `font-style`

Acest atribut definește modul în care este prezentat un anumit font, iar această prezentare poate fi: `normal`, `italic` sau `oblique`. În unele reprezentări ale caracterelor, modul `normal` de scriere este numit și *Roman*. Modul `oblique` de scriere este similar cu modul `italic`, cu

excepția că literele sunt înclinate de către sistem, în loc să se folosească un stil de font diferit, cum ar fi cel italic. Atributul `font-style` se folosește astfel:

```
BODY {font-style: italic}
```

Atributul `font-variant`

La fel ca și atributul `font-style`, acest atribut stabilește scrierea cu minuscule. Cele două valori ale lui sunt: `normal` și `small-caps` (minuscule). Iată un exemplu de folosire a acestui atribut:

```
BODY {font-variant: small-caps}
```

Dacă tipul caracterului nu permite scrierea cu minuscule, sistemul încearcă să reducă majusculele la o dimensiune mai mică pentru a obține litere mici. Ca o ultimă soluție, textul va apărea scris cu toate literele mari.

Atributul `font-weight`

Acest atribut are câteva valori care stabilesc grosimea caracterelor. Valorile principale sunt `normal` și `bold` (aldine). Puteți înlocui aceste valori cu o valoare dintr-o listă de la `100` la `900`. Dacă un tip de caractere are o grosime „medie", valoarea sa corespunde cu `500`. Literele aldine corespund valorii `700`. Iată sintaxa pentru atributul de grosime a fontului.

```
BODY {font-weight: bold}
```

Alte două valori sunt `bolder` (mai îngroșat) și `lighter` (mai subțiat). Acestea măresc grosimea caracterelor cu un nivel peste nivelul curent (de la `200` la `300`) și, respectiv, micșorează grosimea cu un nivel sub nivelul curent (de la `700` la `600`).

Atributul `font-size`

Dimensiunea unui font poate fi definită prin patru metode: dimensiunea absolută, dimensiunea relativă, lungimea sau prin procent.

- **Dimensiunea absolută**: Această metodă se poate implementa în câteva moduri. Primul este cu o valoare care reprezintă dimensiunea fontului în funcție de alte dimensiuni din familia fontului respectiv, `xx-small`, `x-small`, `small` (mici), `medium` (medii), `large` (mari), `x-large`, `xx-large`. Puteți folosi și o valoare numerică, cum ar fi `12pt` (12 puncte), așa cum e prezentat mai jos.

```
BODY {font-size: 18pt}
```

- **Dimensiunea relativă**: Această metodă stabilește dimensiunea în funcție de stilul părinte. Dimensiunea poate fi una dintre valorile `smaller` (mai mici) sau `larger` (mai mari) și variază mărimea în mod crescător sau descrescător, așa cum este prezentat mai jos.

```
P {font-size: smaller}
```

Dacă fontul nu conține o mapare la numele dimensiunilor, se recomandă o scalare cu 1.5 între dimensiuni. De exemplu, un font de 10pt va fi mărit la 15pt și micșorat la 7pt.

- **Lungimea**: Această metodă este o altă variantă a metodei dimensionării relative și constă în stabilirea dimensiunii folosind un factor de scalare a lățimii unui em, de exemplu `1.5 em`, așa cum este prezentat mai jos.

```
P {font-size: 1.5em}
```

- **Procent**: Această metodă oferă de asemenea o specificare relativă care multiplică dimensiunea fontului părinte cu un anumit procent pentru a obține noua dimensiune, cum ar fi de exemplu `150%`, așa cum este exemplificat mai jos.

```
H3 {font-size: 150%}
```

Atributul font

Acest atribut permite configurarea tuturor celorlalte atribute anterioare sub o singură „umbrelă". Ordinea atributelor trebuie să fie, font-syle, font-variant, font-weight, font-size, line-height, font-family. Nu se pun virgule între valorile atributelor, cu excepția familiilor de fonturi.

```
BODY {font: small-caps bold 14pt garamond, palatino, serif}
```

Culoarea și fundalul

Aceste elemente stabilesc culoarea textului (prim plan) și a suprafeței din spatele textului (fundal). Pe lângă stabilirea unei culori de fundal, puteți defini de asemenea și o imagine de fundal. Toate culorile se definesc prin folosirea acelorași opțiuni (numele culorilor sau un triplet hexazecimal), cum ar fi de exemplu în proprietatea color.

Atributul color

Acest atribut definește culoarea elementului text și este precizat prin folosirea unuia dintre cuvintele cheie (cum ar fi red (roșu)). De asemenea, puteți defini culoarea printr-un triplet hexazecimal sau prin valori zecimale separate de virgule, obținând astfel un amestec de roșu, verde și albastru (de exemplu rgb (255, 0, 0)):

```
BLOCKQUOTE {color: rgb(0,255,255)}
```

Atributul background-color

Acest atribut stabilește culoarea fundalului pentru un stil. Puteți stabili acest atribut independent de culoarea fundalului din document pentru a permite evidențierea textului în diverse moduri:

```
BLOCKQUOTE {background-color: blue}
```

Atributul background-image

Acest atribut specifică o imagine de fundal pentru un stil. Îl puteți folosi împreună cu atributul background-color pentru a asigura un efect de înlocuire atunci când imaginea nu mai este disponibilă. Dacă imaginea e disponibilă, ea va fi afișată peste culoarea de fundal. Iată un exemplu de folosire a acestui atribut:

```
BLOCKQUOTE {background-image: url(logo.gif)}
```

Atributul background-repeat

Dacă imaginea din fundal trebuie să se repete (în mozaic), folosiți acest atribut pentru a defini în ce mod. Valorile sale sunt repeat, repeat-x, repeat-y. Valoarea repeat arată că imaginea va fi mozaicată normal. Valoarea repeat-x repetă imaginea pe o singură linie orizontală, iar valoarea repeat-y repetă imaginea pe o singură linie verticală:

```
BLOCKQUOTE {background-image: url(logo.gif);
            background-repeat: repeat-x}
```

Atributul "background-attachment"

Acest atribut, o caracteristică extinsă a imaginilor din fundal, care nu exista anterior în limbajul HTML, specifică dacă imaginea din fundal este atașată la textul din prim plan (scroll) sau ancorată în fundal (fixed). Această caracteristică apare doar atunci când utilizatorul derulează o selecție de text. Iată un exemplu de folosire a acestui atribut:

```
BLOCKQUOTE {background-image: url(logo.gif);
            background-attachment: fixed}
```


Atributul "background-position"

Când folosiți o imagine în fundal, într-un HTML normal, punctul de pornire este plasat întotdeauna în partea din stânga-sus a ecranului. Folosind o foaie de stil de pornire, puteți specifica un punct de pornire oriunde în caseta care conține stilul.

Puteți specifica punctul de pornire al imaginii în trei moduri. Primul mod este prin cuvintele cheie de localizare. Pentru plasarea pe orizontală, opțiunile sunt left (stânga), center (centru) și right (dreapta). Pentru plasarea pe verticală, variantele sunt top (sus), center (centru) și bottom (jos). Alt mod constă în reprezentarea poziției ca un procent din spațiul disponibil, cu 0% 0% fiind în partea din stânga-sus (prestabilită) și 100%, 100% fiind în partea din dreapta-jos. Ultimul mod constă în specificarea coordonatelor în centrimetri sau inci.

Dacă se dă doar o singură valoare pentru plasarea imaginii, ea va fi folosită pentru poziția pe orizontală. Dacă se dau ambele valori, prima valoare e considerată ca fiind pentru orizontală, iar a doua ca fiind pentru verticală.

Iată un exemplu de folosire a cuvintelor cheie pentru localizare pentru a specifica poziția unei imagini de fundal.

```
BLOCKQUOTE {background-image: url(logo.gif);
            background-repeat: repeat-y;
            background-position: right top; }
```

Atributul background

Acest atribut, similar cu atributul font, vă permite definirea unui set de valori pentru fundal într-o singură etapă. Ordinea este background-color, background-image, background-repeat, background-attachment, background-position, așa cum e prezentat în acest exemplu.

```
P { background: black url(logo.gif) repeat-y fixed right top }
```

Textul

Acest set de atribute de stil conține valorile ce pot afecta aspectul unui text, dar nu prin schimbarea directă a tipului de caractere. Acest set include valorile pentru spațiere, subliniere, clipire și tăiere cu o linie. De asemenea, acceptă și unele dintre atributele pentru poziționare, incluzând alinierea la stânga și la dreapta și indentarea.

Atributul word-spacing

Acest atribut, ce indică o suplimentare a spațiului prestabilit dintre cuvinte, are valoarea dată în „em". Un „em" este definit ca fiind spațiul ocupat de litera „m" și este dimensiunea de referință pentru determinarea lățimii unui font. Pentru a aduce valoarea la cea implicită, se folosește 0em sau normal. O spațiere la 1em se realizează astfel.

```
BODY {word-spacing: 1em}
```

Atributul letter-spacing

Acest atribut este similar cu atributul word-spacing, cu excepția că atributul letter-spacing adaugă puțin spațiu suplimentar între litere, așa cum e prezentat mai jos.

```
BODY {letter-spacing: 0.2em}
```

În plus față de metoda pe care o folosește browserul pentru a determina spațierea, distanțarea suplimentară a literelor este afectată și de alinierea textului.

Atributul text-decoration

Acest atribut e înrudit cu atributele din familia font. El stabilește diversele ornamentări aplicate textului, cum ar fi sublinierea, tăierea cu o linie și clipirea. Cele cinci valori ale lui sunt none (nici unul), underline (subliniere), overline (linie peste text), line-through și blink (clipire). Iată un exemplu de folosire a atributului de decorare a textului.

```
STR.blink {text-decoration: underline blink}
```

Atributul vertical-align

Acest atribut stabilește poziționarea pe verticală a textului, fie relativ la o referință absolută, fie în funcție de elementul părinte. El conține o serie de valori și cuvinte cheie:

- **baseline:** Aliniază linia de bază a stilului cu linia de bază a elementului părinte
- **sub:** Atribuie stilul unui indice relativ la elementul părinte
- **super:** Atribuie stilul unui exponent relativ la elementul părinte
- **text-top:** Aliniază partea de sus a textului cu partea de sus a textului părinte
- **text-bottom:** Aliniază partea de jos a textului cu partea de jos a textului părinte
- **middle:** Aliniază punctul de jumătate al elementului pe verticală la linia de bază a părintelui, plus jumătate din înălțimea x a părintelui (înălțimea x este înălțimea literei minuscule x)
- **top:** Aliniază partea de sus a elementului cu cel mai înalt element din linia curentă
- **bottom:** Aliniază partea de jos a elementului cu cel mai mic element din linia curentă.
- **percentage:** Folosind un procent pozitiv sau negativ, ridică sau coboară elementul în funcție de linia de bază a părintelui:

```
SUB { vertical – align: -10% }
```

Atributul text-transform

Acest atribut stabilește scrierea cu majuscule a textului specificat, folosind una dintre cele patru variante: capitalize (prima literă a fiecărui cuvânt), uppercase (toate literele), lowercase (toate literele scrise cu minuscule) și none (nici una). Iată un exemplu de folosire a atributului text-transform:

```
STR.caps { text-transform: uppercase }
```

Atributul text-align

Acest atribut vine în completarea atributului care stabilește alinierea stânga-dreapta-centru din limbajul HTML standard, pentru a asigura o aliniere atât la stânga, cât și la dreapta, după cum este exemplificat mai jos:

```
BLOCKQUOTE { text–align: justify }
```

Dacă browserul nu acceptă alinierea justify, el va fi înlocuit de obicei cu left (stânga).

Atributul text-indent

Acest atribut, definit printr-o valoare absolută în inci sau „em", stabilește dimensiunea spațiului adăugat înainte de prima linie, după cum este exemplificat mai jos:

```
P { text-indent: 5em }
```

Atributul line-height

Acest atribut stabilește distanța dintre liniile de bază adiacente, folosind o lungime (în „em"), un factor de multiplicare sau un procent. Factorii de multiplicare sunt indicați fără unitățile de măsură, cum ar fi 1.5. Când folosiți această metodă, copilul moștenește factorul de multiplicare, nu rezultatul. În exemplul următor, înălțimea liniei devine de 18 puncte, iar mărimea fontului rămâne la 12 puncte:


```
DIV { line-height: 1.5; font-size: 12pt }
```

În acest caz, dacă aveți un paragraf copil al divizării și mărimea prestabilită a textului din paragraf este de 10 puncte, atunci copilul va moșteni factorul de 1.5, făcând dimensiunea de 15 puncte.

Margini, spațieri și borduri

Fiecare element creat într-o foaie de stil este prezentat în propria lui casetă. Deși caseta în sine poate avea propriile caracteristici ce stabilesc modul de conectare a casetei cu elementele din pagină, totuși se vor aplica toate stilurile din element din interiorul casetei. Lungimea este specificată în inci (in), centimetri (cm), „em" (em), puncte (pt) sau pixeli (px).

Caracteristicile casetei se împart în trei categorii principale. Proprietățile de margine stabilesc bordura din exteriorul casetei, proprietățile de spațiere determină cât spațiu trebuie inserat între bordură și conținut, iar proprietățile de spațiere de bordură definesc liniile grafice din jurul unui element.

Alte proprietăți ale casetei mai sunt lățimea, înălțimea și poziția fizică.

Atributele margin-top, margin-bottom, margin-right și margin-left

Aceste patru atribute stabilesc spațiul dintre element și elementele alăturate, indiferent dacă sunt definite prin lungime sau prin procent din lățimea textului părinte sau sunt tratate automat, după cum este exemplificat mai jos.

```
BLOCKQUOTE { margin–top: 4em;
             margin–bottom: auto }
```

Atributul margin

Atributul margin vă furnizează o metodă rapidă de stabilire a dimensiunilor celor patru margini. Dacă specificați cele patru valori, ele sunt aplicate în următoarea ordine : (sus, dreapta, jos, stânga). Dacă specificați doar o valoare, aceasta va fi aplicată tuturor celor patru margini. Dacă specificați două sau trei valori, atunci valorile care lipsesc sunt copiate după valorile marginilor opuse. Iată un exemplu de folosire a atributului margin.

```
BLOCKQUOTE {margin: 4em 2em}
```

Atributele padding-top, padding-bottom, padding-right, padding-left

Aceste atribute stabilesc distanța dintre marginile casetei și elementele din interiorul ei. Distanța poate fi stabilită folosind dimensiunile fizice sau folosind un procent din lățimea părintelui, după cum este exemplificat mai jos :

```
BLOCKQUOTE {padding–top : 110%; padding-bottom: 115%}
```

Atributul padding

Acest atribut furnizează o modalitate rapidă de stabilire a valorilor celor patru spații prezentate mai sus. Când specificați cele patru valori, ele sunt aplicate în următoarea ordine: (sus, dreapta, jos, stânga). Dacă specificați doar o valoare, aceasta va fi aplicată tuturor celor patru. Dacă specificați două sau trei valori, atunci valorile care lipsesc, sunt copiate după valorile marginilor opuse. Iată un exemplu de folosire a atributului padding.

```
BLOCKQUOTE {padding: 10pt 12pt}
```

Atributele border-top, border-bottom, border-right și border-left

Aceste patru atribute stabilesc stilul și culoarea fiecărei borduri din jurul unui element. Stilurile se definesc folosind unul dintre cuvintele cheie pentru stilul bordurii: none (nici una), dotted (linie punctată), dashed (linie întreruptă), solid (linie continuă), double (linie

dublă), groove (tridimensional brăzdată), ridge (tridimensional stil creastă), inset (tridimensional înăuntru) și outset (tridimensional în afară). Pentru mai multe informații privind aceste cuvinte cheie, parcurgeți informațiile despre border-style, din acest capitol.

Culorile se definesc prin folosirea unui cuvânt cheie pentru fiecare culoare. Pentru mai multe informații, consultați secțiunea despre border-color din acest capitol.

```
BLOCKQUOTE {border-left: solid red}
```

Atributele border-top-width, border-bottom-width, border-right-width și border-left-width

Aceste atribute definesc o bordură fizică în jurul unei casete, similară cu bordura folosită pentru tabelele din HTML. În plus, pe lângă definirea unei anumite lățimi în „ems", puncte sau inci, puteți folosi și cuvintele cheie thin (subțire), medium (mediu) și thick (gros). Folosirea unităților de măsură în „em" conduce la obținerea unei borduri a cărei lățimi se schimbă în funcție de dimensiunea fontului curent. Iată un exemplu de stabilire a atributelor border.

```
STR  {border-right-width: 2 pt;
      border-left-width: 2pt}
```

Atributul border-width

Atributul border-width furnizează o metodă rapidă pentru stabilirea lățimii celor patru borduri din jurul unei casete. Când precizați cele patru valori, ele sunt aplicate, în ordine, marginilor de sus, dreapta, jos și stânga. Dacă precizați doar o valoare, ea va fi aplicată tuturor marginilor. Dacă folosiți două sau trei valori, valorile care lipsesc sunt copiate din valorile marginilor opuse lor în casetă.

```
BLOCKQUOTE {border-width: medium 0pt 0pt thick}
```

Atributul border-color

Acest atribut stabilește culoarea pentru toate cele patru margini ale bordurii și folosește pentru valoarea ei un singur cuvânt cheie. Puteți stabili culoarea fiecărei margini separat. Iată un exemplu de folosire a atributului border-color.

```
BLOCKQUOTE {border-color: yellow}
```

Atributul border-style

Aspectul bordurii poate fi stabilit prin diverse valori: none (nici unul), dotted (linie punctată), dashed (linie întreruptă), solid (linie plină), double (linie dublă), groove (tridimensional brăzdată), ridge (tridimensional stil creastă), inset (tridimensional înăuntru) și outset (tridimensional în afară). Ultimele patru valori sunt reprezentate tridimensional (3D), dacă browserul acceptă această reprezentare sau browserul poate reprezenta toate valorile cu linie plină, cu excepția valorii none.

La fel ca și atributul border-color, stilul este aplicat uniform tuturor celor patru margini. Atributul este definit astfel:

```
BLOCKQUOTE {border-style: groove}
```

Atributul border

Atributul border furnizează o metodă rapidă pentru stabilirea tuturor variabilelor bordurii, incluzând lățimea, stilul și culoarea, după cum este prezentat mai jos.

```
BLOCKQUOTE {border: 1.5pt double black}
```


Stabilește valorile pentru toate cele patru margini în același timp, fiind prioritar oricăror definiri individuale făcute anterior, pentru același element.

Atributul height

Acest atribut stabilește înălțimea totală a casetei care conține fie elementul text, fie elementul imagine. Când conținutul este de tip text, se adaugă bare de derulare după cum este necesar pentru ca tot materialul să fie disponibil pentru utilizator. Când conținutul este de tip imagine, atunci este scalat pentru a se potrivi în interiorul suprafeței. Puteți stabili o valoare fizică, după cum este exemplificat mai jos, sau puteți folosi elementul auto pentru a lăsa browserul să aloce el spațiul necesar.

```
BLOCKQUOTE {height: 100px}
```

Atributul width

Similar cu atributul height, atributul width stabilește lățimea totală a casetei care conține elementul. Când conținutul casetei este de tip text, se adaugă bare de derulare după cum este necesar pentru a face materialul disponibil pentru utilizator. Dacă sunt folosite și elemente cu o imagine, iar valoarea unui element este auto, după cum este prezentat mai jos, raportul între lățimea și înălțimea imaginii este menținut.

```
BLOCKQUOTE {width: auto}
```

Atributul float

Acest atribut stabilește o valoare similară cu valoarea atributului align folosit în HTML. Cele trei valori posibile sunt left (stânga), right (dreapta) și none (nici una). Valoarea none permite ca elementul să fie plasat acolo unde este posibil, iar celelalte două valori forțează plasarea elementului în stânga sau în dreapta ecranului, textul urmând să fie plasat în partea opusă, în jurul elementului. Iată un exemplu de stabilire a atributului float pentru un element BLOCKQUOTE:

```
BLOCKQUOTE {float: right}
```

Atributul clear

Acest atribut este similar cu atributul clear, folosit cu eticheta <BR>, din HTML și folosește aceleași cuvinte cheie ca și atributul float. Dacă îl folosiți cu right sau left, elementele se vor muta sub oricare element flotant, pe partea respectivă, așa cum e prezentat mai jos:

```
BLOCKQUOTE {clear: left right}
```

Dacă stabiliți la valoarea none, elementele flotante sunt plasate pe ambele părți.

Clasificare

Aceste atribute controlează comportarea generală a celorlalte elemente, mai mult decât să definească doar aspectul. În plus, clasificarea include atributele pentru articolele de listă, identificate în limbajul HTML cu eticheta <LI>.

Atributul display

Acest atribut precizează dacă și când trebuie să fie folosit un element de stil. Comportarea lui este determinată de patru cuvinte cheie:

- **inline:** Creează o nouă casetă pe aceeași linie cu elementele de text alăturate și este formatată în funcție de mărimea și cantitatea de conținut aflat între marginile sale, cum ar fi o imagine (IMG) sau text (STR).
- **block:** Creează o nouă casetă în funcție de elementele înconjurătoare. Acesta este folosit uzual cu elemente cum ar fi H1 și P.

- **list-item:** Este similar cu `block`, cu deosebirea că sunt adăugate marcatoare de articole din listă, care se comportă mai mult ca un conținut inline.
- **none:** Oprește afișarea elementului în orice situație, inclusiv pentru copil sau element.

Următorul exemplu arată cum sunt folosite cuvintele cheie `inline` și `block`.

```
IMG {display: inline}
BLOCKQUOTE {display: block}
```

Atributul white-space

Numele acestui atribut poate conduce la o interpretare greșită deoarece se referă la modul cum sunt tratate spațiile și întreruperile de linie. Variantele sunt `normal` (unde spațiile suplimentare sunt ignorate), `pre` (la fel ca și în textul preformatat din HTML) și `nowrap` (la care liniile sunt întrerupte numai cu atributul <BR>). Iată un exemplu de folosire a acestui atribut:

```
BLOCKQUOTE {white -space: pre}
```

Atributul list-style-type

Acest atribut configurează o listă. Variantele sunt `disc` (disc), `circle` (cerc), `square` (pătrat), `decimal` (zecimal), `lower-roman` (cifre romane mici), `upper-roman` (cifre romane mari), `lower-alpha` (litere mici), `upper-alpha` (litere mari) și `none` (nici una). Pentru mai multe informații despre cum sunt reprezentate fiecare dintre acestea pe ecran, consultați Capitolul 8, „Folosirea listelor pentru organizarea informațiilor". În continuare sunt prezentate câteva exemple de folosire a acestui atribut:

```
LI.outline1 {list-syle-type: upper-roman}
LI.outline2 {list-syle-type: upper-alpha}
LI.outline3 {list-syle-type: decimal}
```

Atributul list-style-image

În locul unui marcator de text pentru articolul de listă, puteți specifica și adresa URL a unei imagini pe care să o folosiți. Dacă imaginea nu este disponibilă, este folosit în mod implicit marcatorul de text. Acest atribut poate fi folosit astfel:

```
LI.general {list-syle-image: url(bullet.jpg)}
```

Atributul list-style-position

Cele două valori ale acestui atribut, `inside` (interior) și `outside` (exterior) determină formatarea textului care urmează după marcatorul articolului de listă. Valoarea `outside`, valoare prestabilită, aliniază liniile suplimentare ale textului sub prima linie, cu primul caracter din prima linie. Dacă folosiți valoarea `inside`, așa cum este prezentat în exemplul următor, a doua linie și liniile următoare sunt aliniate la marcatorul de articol din listă.

```
LI {list-syle-position: inside}
```

Atributul list-style

Această proprietate furnizează o metodă rapidă pentru folosirea atributelor `list-style-type`, `list-style-image` și `list-style-position`, așa cum e prezentat în exemplele următoare:

```
OL {list-syle: lower-alpha outside}
UL {list-syle: square url (bullet.jpg) inside}
```

Anexa D
Scurtă prezentare a limbajului XML

de Simon North

În această anexă

- **Sintaxa limbajului XML** 827
- **Legăturile în limbajul XML** 832

Această anexă este o versiune foarte redusă a celor două standarde care guvernează limbajul XML („Partea întâi: Sintaxa și „Partea a 2-a: Legături") care deja au fost publicate oficial.

Această versiune despre specificații nu are nici o autoritate și nu are intenția să înlocuiască documentațiile oficiale. Este destinată să vă ofere o scurtă prezentare a regulilor esențiale din specificații, fără reguli de gramatică sau sintaxă și fără clarificări sau exemple. Aceste părți ale standardelor sunt foarte folositoare când întâlniți limbajul XML pentru prima oară, dar cu timpul, gradul lor de detaliere poate deveni un obstacol în găsirea unor informații. Prin eliminarea explicațiilor introductive și a exemplelor, sperăm că această scurtă prezentare vă va fi o referință de durată prin faptul ca vă va permite să vă concentrați asupra elementelor esențiale care trebuie să fie reținute.

Titlurile de secțiuni originale au fost păstrate neschimbate, pentru a le putea găsi rapid în documentația originală.

Sintaxa limbajului XML

Următoarele secțiuni concentrează specificația despre sintaxa XML.

Documente

O secțiune de text este considerată un document XML, dacă este fie valabilă fie bine formulată.

Structura logică și fizică

Fiecare document XML are o structură logică, care conține *declarații, elemente, comentarii, informații despre caractere* și *instrucțiuni de procesare*, toate indicate explicit de către o structură fizică și de marcare alcătuită din *entități*.

Documente XML bine formulate

O secțiune de text poate fi considerată un document XML bine formulat dacă îndeplinește următoarele cerințe:

- Conține unul sau mai multe elemente.
- Îndeplinește toate cerințele pentru o bună formulare.
- Toate elementele sunt imbricate unele în altele.

Caractere

Datele memorate într-o entitate XML sunt reprezentate fie în format text, fie în format binar. Datele binare au o notație asociată, identificată prin nume.

Comentarii

Comentariile pot fi plasate oriunde, cu excepția secțiunii CDATA, declarațiilor și etichetelor. Șirul de caractere - - (două liniuțe de despărțire în silabe) nu trebuie să apară în interiorul comentariilor.

Instrucțiuni de procesare

Instrucțiunile de procesare al căror nume începe cu literele XML sunt rezervate.

Secțiunile CDATA

- Secțiunile CDATA pot apărea oriunde pot apărea datele de tip caracter.
- Caracterele (<) și ampersandul (&) pot apărea în forma lor literală.
- Secțiunile CDATA nu pot fi imbricate.

Tratarea spațiului liber

Atributul XML-SPACE identifică elementele în care spațiul liber trebuie să fie tratat aparte de către aplicații.

- **DEFAULT:** Înseamnă că procesarea prestabilită a spațiului liber este acceptabilă.
- **PRESERVE:** Înseamnă că toate spațiile libere trebuie să fie menținute.

Prologul și declarația tipului de document

- Documentele XML ar trebui să înceapă cu o declarație XML care să precizeze versiunea limbajului XML folosit.
- Declararea tipului de document trebuie să fie plasată înainte de prima etichetă de început din document.
- Declararea tipului de document XML poate include un indicator către o entitate externă, ce conține o submulțime formată din declarațiile de marcare necesare și poate, de asemenea, să includă o altă submulțime internă. Aceste două submulțimi alcătuiesc definirea tipului de document (DTD) și nici o sumulțime nu poate fi goală.
- Dacă sunt folosite atât submulțimile interne cât și cele externe, procesorul XML trebuie să citească mai întâi submulțimea internă. Declarațiile de atribute și entități din submulțimea internă au întâietate față de cele din submulțimea externă.
- Numele din DTD trebuie să se potrivească cu tipul de element al elementului rădăcină.

Declarația de marcare cerută

Declarația de marcare cerută (RMD), precizează dacă este necesară procesarea DTD:

- **NONE:** Procesorul XML poate analiza corect documentul inclus fără să citească inițial nici o parte din DTD.
- **INTERNAL:** Procesorul XML trebuie să citească și să proceseze submulțimea internă a definiției DTD, dacă este dată.
- **ALL (prestabilită):** Procesorul XML trebuie să citească și să proceseze declarațiile din ambele submulțimi ale definiției DTD, dacă sunt date.

RMD trebuie să arate că este cerută întreaga definiție DTD, dacă submulțimea externă conține atribute nespecificate, care au valori prestabilite sau referințe despre entități, sau dacă spațiul liber apare în interiorul tipurilor de element care conțin elemente.

Structuri logice

- Un document XML conține unul sau mai multe elemente, care sunt delimitate fie de etichete de început și sfârșit fie, pentru elementele goale, doar de etichetele de început.
- Fiecare element are un anumit tip, identificat prin nume, și poate conține un set de atribute.
- Fiecare atribut are un nume și o valoare.

Etichetele de început și de sfârșit

- Începutul fiecărui element XML este marcat de o etichetă de început.
- Sfârșitul fiecărui element este marcat de o etichetă de sfârșit care conține același nume care se regăsește în eticheta de început.
- Dacă un element este gol, el trebuie să fie încheiat cu o etichetă de sfârșit sau eticheta de început reprezintă întregul element, iar eticheta va avea o formulă specială: `<name/>`.
- Fiecare tip de element folosit trebuie declarat. Conținutul unui exemplu de element trebuie să fie identic cu modelul de conținut declarat pentru acel tip de element.

Declarațiile de elemente

- Structura unui element de document XML poate fi limitată, în scopul validării, de folosirea declarațiilor de atribute și elemente.
- O declarație de element impune tipul elementului și conținutul lui. Într-un document XML valid, nici un tip de element nu poate fi declarat mai mult de o singură dată.
- Conținutul unui element poate fi de tip *element content* (conținând doar alte elemente) sau *mixed content* (conținând date de tip caracter combinate cu elemente).
- Un element declarat cu ajutorul cuvântului cheie EMPTY trebuie să fie gol atunci când apare în document.
- Un element declarat cu ajutorul cuvântului cheie ANY, poate conține oricâte elemente copil de orice tip, combinate cu date de tip caracter.

Declarațiile listei de atribute

- Atributele pot apărea doar în etichetele de început.
- Declarațiile listei de atribute precizează numele, tipul datei și valoarea prestabilită (dacă există) a fiecărui atribut asociat cu un tip de element dat.
- Atunci când se precizează mai mult de o singură declarație cu lista de atribute, pentru un tip de element dat, toate conținuturile sunt combinate.
- Atunci când se precizează mai mult decât o singură definiție pentru același atribut al unui tip de element dat, este luată în considerare prima declarație, iar declarațiile care urmează sunt ignorate.

Tipurile de atribute

- Tipurile de atribute în limbajul XML sunt de trei feluri: tip șir, un set de tipuri simbolizate și tip enumerat.
- Tipul șir poate avea ca valoare orice șir de litere.
- Tipurile simbolizate au diferite restricții semantice și lexicale:

  Un tip ID trebuie să fie un simbol NAME valid. Numele trebuie să apară doar o singură dată ca un ID, într-un document XML.

  Un IDREF trebuie să se potrivească cu valoarea unui atribut ID dintr-un element din documentul XML.

  Valorile numelor de entități de acest tip trebuie să se potrivească exact cu numele unei entități generale externe în binar, declarată în DTD.

Tipul nume de simbol trebuie să conțină un șir care să se potrivească cu tipul NMTOKEN (o literă sau o subliniere, urmată de una sau mai multe litere sau digiți.)

Numele notațiilor de atribute trebuie să se potrivească cu unul dintre numele notațiilor incluse în declarație.

- Valorile enumerărilor trebuie să se potrivească cu unul dintre simbolurile NMTOKEN din declarație. Același NMTOKEN nu ar trebui să apară mai mult de o dată în tipurile de atribute enumerate ale unui singur tip de element.

Valorile prestabilite ale atributelor

O declarație de atribut furnizează informații referitoare la obligativitatea prezenței atributului și, dacă nu este obligatorie, cum trebuie să reacționeze un procesor XML dacă un atribut declarat lipsește din document.

- **# FIXED:** Documentul este invalid dacă atributul are o valoare diferită de valoarea prestabilită.
- **# REQUIRED:** Documentul este invalid dacă există o etichetă de început pentru tipul de element, ce nu specifică o valoare pentru atribut.
- **# IMPLIED:** Dacă atributul este omis, procesorul XML trebuie pur și simplu să informeze aplicația că nu a fost specificată nici o valoare.

Dacă atributul nu are nici valoarea #REQUIRED nici #IMPLIED, valoarea conține valoarea prestabilită declarată. Dacă se declară o valoare prestabilită și atributul este omis, atributul este tratat ca și cum ar fi prezent, iar valoarea sa este valoarea prestabilită declarată.

Secțiunile condiționale

Secțiunile condiționale sunt porțiuni din submulțimea externă DTD, care sunt incluse sau excluse din structura logică a definiției DTD, conform cuvintelor cheie care le controlează. O secțiune condițională, poate conține una sau mai multe declarații complete, comentarii, instrucțiuni de procesare sau secțiuni condiționale imbricate, cu spații libere intercalate.

Când cuvântul cheie al secțiunii condiționale este INCLUDE, secțiunea condițională este citită și procesată normal. Când cuvântul cheie este IGNORE, delcarațiile din secțiunea condițională sunt ignorate. Dacă o secțiune condițională cu un cuvânt cheie INCLUDE apare în interiorul unei secțiuni condiționale mai mari, cu un cuvânt cheie IGNORE, atât secțiunea condițională internă cât și cea exterioară sunt ignorate.

Când cuvântul cheie al secțiunii condiționale este o referință a unei entități parametru, entitatea parametru este înlocuită cu valoarea sa înainte ca procesorul să decidă dacă să includă sau să ignore secțiunea condițională.

Structuri fizice

Un document XML are o singură entitate, numită *document entity* (entitatea documentului), care servește ca punct de pornire pentru procesorul XML (și poate conține întregul document).

Entitățile pot fi în format binar sau text. O *entitate text* conține date text, ce sunt considerate ca fiind parte integrantă din document. O *entitate binară* conține date binare cu o notație asociată. Doar entitățile text pot fi menționate folosind referințele de entități și doar numele entităților binare pot fi date ca valoare pentru atributele entități.

Structuri logice și fizice

Structurile logice și fizice (elementele și entitățile) dintr-un document XML trebuie să fie sincrone. Etichetele și elementele trebuie să înceapă și să se termine fiecare în aceeași entitate, dar se pot referi intern la alte entități. Comentariile, instrucțiunile de procesare,

referințele de caractere și referințele de entități trebuie fiecare să fie conținute în totalitate într-o singură entitate.

Entitățile trebuie fiecare să conțină un număr întreg de elemente, comentarii, instrucțiuni de procesare și referințe, pe cât posibil cu date de tip caracter care nu se mai află în alt element din entitate, altfel ele trebuie să conțină date netextuale și nici un element.

Referințele de caracter și entități

Pentru documentele XML bine formulate se aplică următoarele reguli referințelor de caracter și entități:

- Numele dat în referința entitate trebuie să se potrivească exact cu numele dat în declarația entității.
- În documentele XML bine-formulate nu trebuie să declarați entitățile amp, lt, gt, apos și quot. Totuși, în documentele valabile, aceste entități trebuie să fie declarate. Declararea entităților parametri trebuie să preceadă referința.
- O referință entitate nu trebuie să conțină numele unei entități binare. Vă puteți referi la entitățile binare numai când aveți atribute de tip ENTITY sau ENTITIES.
- O entitate de tip text sau parametru nu trebuie să conțină o referință directă sau indirectă spre ea însăși.

Declararea entităților

Dacă aceeași entitate este declarată de mai multe ori, prima declarație întâlnită este cea care se leagă.

Entități interne

Dacă definirea unei entități este doar o valoare, ea se numește *entitate internă*, iar textul ei de înlocuire este dat în declarație. Entitatea parametru și referințele de caracter sunt recunoscute în interiorul valorii entității și sunt extinse imediat. Referințele generale de entitate, din textul de înlocuire, nu sunt recunoscute când este analizată declarația de entitate, deși ele pot fi recunoscute când se face referire la entitatea propriu-zisă.

Entități externe

- Dacă entitatea nu este internă, atunci ea este o entitate externă.
- Într-un document XML valabil, numele trebuie să se potrivească cu numele declarat al unei notații.
- Cuvântul cheie SYSTEM definește o adresă URL, care poate fi folosită pentru a regăsi entitatea.
- În plus, un identificator extern poate să includă cuvântul cheie PUBLIC și un identificator public. Procesorul XML ar putea încerca să genereze o adresă URL alternativă din acest identificator, dar dacă nu poate, trebuie să folosească adresa URL specificată de către cuvântul cheie SYSTEM.

Codificarea caracterelor în entități

Fiecare entitate text externă dintr-un document XML poate folosi o codificare diferită pentru caracterele sale. XML furnizează o instrucțiune de procesare a declarației de codificare, care, dacă apare, trebuie să fie la începutul unei entități sistem, înainte de orice altă dată de tip caracter sau marcaj. În entitatea document, declarația de codificare este parte din declarația XML; în alte entități este parte din instrucțiunea de procesare a codificării.

Entitatea document

Entitatea document servește ca rădăcină a arborelui de entități și ca punct de pornire pentru un procesor XML.

Modul în care procesorul XML tratează entitățile

Atunci când un procesor XML întâlnește caractere și referințe generale de entități sau numele unei entități binare externe ca fiind valoarea unui atribut ENTITY sau ENTITIES, se aplică următoarele reguli.

- Informează întotdeauna aplicația despre apariția referinței și identificatorul ei.
- Trebuie să înlăture referințele la entități și caractere din datele de text înainte de a trece datele către aplicație.
- Trebuie să trimită aplicației caracterul indicat în locul unei referințe de caracter.
- Trebuie să informeze aplicația de existența unui sistem sau identificator public, al entității externe dacă există.
- Trebuie să informeze aplicația de existența unui nume de notație asociat entității binare externe și de sistemul și identificatorul public asociați notației, dacă există.
- Trebuie să includă o entitate (text) internă și să o proceseze ca parte din document.
- Pentru a valida un document XML, procesorul trebuie să includă conținutul unei entități text externe.
- Dacă nu încearcă să valideze un document XML, procesorul poate include conținutul unei entități text externe, dar nu este obligatoriu.

Referințele despre entitățile parametru pot fi folosite în diverse locuri în DTD. Ele sunt expandate imediat ce sunt recunoscute și DTD trebuie să satisfacă regulile de gramatică relevante, după ce ele au fost toate expandate.

Entități predefinite

- Caracterul „&" și semnul „<" trebuie să fie incluse ca secvențe „escape" prin folosirea fie a referințelor de caracter numerice, fie a șirurilor `&` și `<`.
- Caracterul „>" poate fi reprezentat prin folosirea șirului `>`. Apostroful „ ' " poate fi reprezentat prin `'` iar ghilimelele (") pot fi reprezentate prin `"`.

Toate procesoarele XML trebuie să recunoască aceste entități indiferent dacă ele sunt declarate sau nu. Documentele XML valabile trebuie să declare aceste entități înainte de a le folosi.

Declarații pentru notații

Notațiile identifică prin nume formatul entităților binare externe. Declarațiile pentru notatii furnizează un nume pentru notație, pentru folosirea în entitate, în declarațiile listei de atribute și în specificațiile cu valorile atributelor. Ele furnizează de asemenea și un identificator extern pentru notația ce permite unui procesor XML sau aplicației sale client să găsească o aplicație ajutătoare capabilă să proceseze datele din notația dată.

Conformarea

Procesoarele XML de confirmare se împart în două clase: de validare și de ne-validare. Sistemele de validare și ne-validare trebuie să raporteze încălcările cerințelor de bună formulare, date în specificații.

Procesoarele de validare trebuie să raporteze locațiile unde documentul nu se supune cerințelor exprimate în declarațiile din DTD. Ele trebuie de asemenea să raporteze toate încercările nereușite de satisfacere a cerințelor de validare.

Legăturile în limbajul XML

Următoarele secțiuni concentrează specificația despre legături în XML, care prezintă o metodă de descriere a legăturilor dintre obiecte și secțiuni de adresare ale documentelor XML.

Recunoașterea legăturii

Existența unei legături este afirmată de către un element de legătură. Elementele de legătură din limbajul XML se recunosc prin XML-LINK. Valorile posibile sunt SIMPLE, EXTENDED, LOCATOR, GROUP și DOCUMENT. Acestea semnalează că elementul urmează să fie tratat ca fiind de acel tip.

Remaparea atributelor

Pentru a evita conflictele cu elementele ale căror atribute au același nume, atributele pot fi declarate ca echivalente, folosind atributul XML-ATTRIBUTES.

Acest atribut trebuie să conțină perechi de nume. În fiecare pereche, primul nume trebuie să fie ROLE, HREF, TITLE, SHOW, INLINE, CONTENT-ROLE, CONTENT-TITLE, ACTUATE, BEHAVIOR sau STEPS. Al doilea nume este tratat ca și cum el ar juca rolul atribuit primului.

Pentru a evita declararea explicită a atributelor pentru fiecare element, pot fi declarate valori prestabilite pentru atribute.

Elemente de legătură

Legăturile simple sunt de obicei de tip inline și întotdeauna sunt uni-direcționale. Legăturile extinse pot fi de tip inline (în documentul sursă) sau out-of-line (în afara documentului sursă) și pot fi folosite pentru legături multidirecționale, legături spre date de tip „read-only" (protejate la scriere) etc.

Informații asociate cu legăturile

Informațiile pot fi asociate cu legăturile folosind următoarele atribute:

- **ROLE:** Este un șir care identifică sensul unei legături la un program de aplicație. Fiecărui participant la o legătură i se poate atribui rolul său propriu.
- **HREF:** Identifică ținta.
- **TITLE:** Denumește legătura.
- **SHOW și ACTUATE:** Arată comportarea elementului în timp ce urmărește legătura.
- **INLINE:** Specifică dacă elementul de legătură este inline.

Conținutul elementelor de legătură

Orice element poate fi recunoscut ca element de legătură în funcție de folosirea atributului XML-LINK. Într-un document XML valid, fiecare element de legătură trebuie să se conformeze cerințelor specificate în definiția DTD care se aplică.

Legături simple

O legătură simplă poate conține doar o singură adresă; atributele adresă sunt atașate direct elementului de legătură.

Legături extinse

O legătură extinsă poate implica oricâte resurse, dar nu are nevoie să fie co-localizată cu nici una din ele. O aplicație poate urma legăturile prin toate resursele. Adresele unei legături extinse sunt memorate în elementele copil ale elementului de legătură, fiecare cu setul ei de atribute.

Rețineți că multe dintre atribute pot fi furnizate atât pentru elementul de legătură părinte, cât și pentru elementul adresă copil. Dacă un astfel de atribut este introdus în elementul de legătură, dar nu este introdus în elementul adresă, valoarea dată în elementul de legătură va fi folosită la procesarea elementului adresă (locator).

Legături de tip inline și out-of-line

Atributul INLINE poate lua valorile TRUE (adevărat) și FALSE (fals). Valoarea TRUE, care este valoare prestabilită, arată că întreg conținutul elementului de legătură este considerat ca fiind parte integrantă din legătură, cu excepția elementelor adresă copil (care sunt considerate ca fiind parte integrantă a mecanismului elementului de legătură).

Când legătura este de tip inline, atributele CONTENT-ROLE și CONTENT-TITLE, pot fi folosite pentru a furniza informațiile referitoare la titlu și rol pentru această resursă „conținut". Dacă INLINE are valoarea FALSE, atributele CONTENT-TITLE și CONTENT-ROLE nu au nici un efect.

Comportarea legăturii

Cronometrarea și efectele de urmărire a unei legături pot fi reprezentate folosind atributele SHOW și ACTUATE.

SHOW

Atributul SHOW arată cum trebuie să fie afișată sau procesată o țintă atunci când se urmărește o legătură:

- **EMBED:** ținta trebuie să fie înglobată în corpul sursei de unde a pornit legătura.
- **REPLACE:** ținta trebuie să înlocuiască sursa de unde a pornit legătura.
- **NEW:** ținta trebuie să fie afișată sau procesată într-un context nou, fără să-l afecteze pe cel de unde a pornit legătura.

ACTUATE

Atributul ACTUATE precizează când trebuie să fie urmată o legătură:

- **AUTO:** Când acest atribut este întâlnit și afișarea sau procesarea sursei nu este considerată completă, până ce nu se realizează legătura.
- **USER:** Nu se urmează legătura până ce nu apare o cerere externă explicită de urmare a legăturii.

Adresare

O adresă întotdeauna conține o adresă URL. Când o adresă identifică o țintă care este un document XML, valoarea adresei poate conține o adresă URL, un identificator de fragment sau pe amândouă.

Indicatori extinși

Indicatorii extinși operează pe arborele de elemente definit de elementele din documentul XML. Forma de bază este o serie de termeni de locație, fiecare precizând fie o locație absolută, fie una relativă. Fiecare termen are un cuvânt cheie și poate fi calificat de anumiți parametri, cum ar fi număr de exemplu, un tip de element sau atribut.

Termeni pentru locații absolute

Locațiile absolute sunt specificate prin folosirea unuia dintre următoarele cuvinte cheie.

ROOT

Cuvântul cheie ROOT precizează că sursa locației este elementul rădăcină al sursei. Aceasta este comportarea prestabilită.

HERE

Cuvantul cheie HERE precizează că sursa locației pentru primul ei termen de localizare a seriei respective este elementul de legătură care conține adresa și nu elementul rădăcină prestabilit.

DITTO

Cuvântul cheie DITTO precizează că sursa locației pentru primul ei termen de localizare este cea specificată de întreg primul indicator.

ID

Cuvântul cheie ID precizează că sursa locației pentru primul termen de localizare este elementul din sursă care conține un atribut de tip ID cu o valoare care se potrivește cu numele dat.

HTML

Cuvântul cheie HTML selectează primul element al cărui tip este A și care are un atribut NAME cu aceeași valoare ca și valoarea NAME furnizată (este similar cu caracterul # dintr-un document HTML).

Termeni pentru locații relative

Locațiile relative sunt definite de unul dintre următoarele cuvinte cheie.

- **CHILD:** Selectează elementele copil ale sursei locației.
- **DESCENDANT:** Selectează elementele care apar în contextul sursei locației.
- **ANCESTOR:** Selectează elementele în care este găsită sursa locației.
- **PRECEDING:** Selectează elementele care apar înainte de sursa locației.
- **PSIBLING:** Selectează elementele frate (sibling) precedente ale sursei locației.
- **FOLLOWING:** Selectează elementele care apar după sursa locației.
- **FSIBLING:** Selectează elementele frate care apar după sursa locației.

Candidații pot fi selectați după numărul de apariții, tipul elementului, numele atributului și valoarea atributului.

Grupurile de legături extinse

Un grup de legături extinse este folosit pentru a memora o listă de legături către alte documente ce formează un grup de documente interconectate.

Fiecare document este identificat după atributul HREF al unui element document de legătură extinsă, care este un element copil din GROUP. Valoarea unui atribut HREF este o adresă.

Aceste elemente sunt identificate prin folosirea atributului XML-LINK împreună cu valoarea GROUP sau DOCUMENT.

Anexa E
JAVASCRIPT

Prima parte a acestei anexe conține obiectele, cu proprietățile și metodele prezentate împreună cu obiectul la care se aplică. A doua parte a acestei anexe prezintă funcțiile independente din JavaScript, care nu sunt corelate cu un anumit obiect, precum și operatorii din JavaScript.

O observație despre JavaScript 1.2

JavaScript 1.2 este proiectat pentru a asigura o interfață lină cu Netscape Navigator 4.0. Au fost introduse noi caracteristici în diverse secțiuni ale modelului de limbaj, incluzând următoarele fără a ne limita aici:

- Evenimente
- Obiecte
- Proprietăți
- Metode

Browserul Netscape Navigator 4.0 a fost scris pentru a accepta aceste noi caracteristici, dar nu și versiunile mai vechi de Navigator. De aceea compatibilitatea cu versiunile mai vechi este o problemă.

În această anexă, tehnicile care funcționează doar cu browserul Netscape Navigator 4.0 sau cu versiunile mai noi sunt marcate explicit prin precizarea în fiecare titlu a cuvintelor „Doar Navigator 4.0".

În final, trebuie reținut că la scrierea în JavaScript trebuie precizată clar care versiune de JavaScript este folosită. Dacă nu se specifică, scripturile dumneavoastră s-ar putea să nu funcționeze. Versiunea se identifică prin folosirea atributului LANGUAGE din eticheta <SCRIPT>. Iată câteva exemple:

```
<Script Language = "JavaScript"> - Compatibil cu 2.0 sau mai nou

<Script Language = "JavaScript 1.1"> - Compatibil cu 3.0 sau mai nou

<Script Language = "JavaScript 1.2"> - Compatibil cu 4.0 sau mai nou
```

Pentru a indica unde sunt implementate obiectele, metodele, proprietățile și elementele de tratare a evenimentelor, sunt folosite următoarele coduri după titlurile secțiunilor:

- C: Client JavaScript (Serverul JavaScript nu este prezentat în această anexă)
- 2: Netscape Navigator 2
- 3: Netscape Navigator 3
- 4: Doar Netscape Navigator 4 (aceasta nu înseamnă că Netscape Navigator 4 funcționează doar cu aceste articole; el va trata toate implementările).
- I: Microsoft Internet Explorer 3

Obiectul anchor [C | 2 | 3 | 4 | I]

Obiectul anchor definește o ancoră HTML.

Proprietăți

- **name:** Reprezintă un șir care indică numele ancorei. [Nu 2|3].

Obiectul applet [C | 3]

Obiectul applet reprezintă o miniaplicație Java introdusă într-o pagină Web cu eticheta <APPLET>.

Proprietăți

- **name:** Un șir care reprezintă atributul <NAME> al etichetei <APPLET>.

Obiectul area [C | 3]

Obiectul area delimitează o zonă pe care se poate efectua clic, definită într-o hartă imagine. Obiectele area apar ca intrări în tabloul links (legături) al obiectului document.

Proprietăți

- **hash:** Un șir care indică un nume de ancoră din adresa URL.
- **host:** Un șir care indică gazda și secțiunea cu numele domeniului din adresa URL.
- **hostname:** Un șir care indică gazda, numele de domeniu și numărul portului din adresa URL.
- **href:** Un șir care reprezintă întreaga adresă URL.
- **pathname:** Un șir care indică secțiunea de cale din adresa URL (exclusiv gazda, numele domeniului, numărul portului și protocolul).
- **port:** Un șir care indică numărul portului din adresa URL.
- **protocol:** Un șir care indică secțiunea din adresa URL rezervată protocolului, inclusiv cele două puncte finale.
- **search:** Un șir care indică secțiunea de interogare din adresa URL (după semnul de întrebare).
- **target:** Un șir care indică atributul TARGET din eticheta <AREA>.

Metode

- **getSelection:** Prin această metodă se obține selecția curentă și se returnează această valoare sub forma unui șir.

Elementele de tratare a evenimentelor

- **onDblClick:** Precizează codului scris în JavaScript că trebuie să fie executat atunci când utilizatorul efectuează un dublu clic pe zona sa. Doar pentru Netscape Navigator 4.0. (Nu este implementat pe Macintosh). [4]
- **onMouseOut:** Precizează codului scris în JavaScript că trebuie să fie executat atunci când mouse-ul este mutat în afara zonei specificate de eticheta <AREA>.

Proprietăți noi în JavaScript 1.2

type	Indică un eveniment MouseOut.
target	Indică obiectul la care a fost trimis evenimentul.
layer[n]	Unde [n] reprezintă X sau Y, folosite (împreună cu page[n] și screen[n] pentru a descrie poziția cursorului unde a apărut evenimentul MouseOut.
page[n]	Unde [n] reprezintă X sau Y, folosite (împreună cu layer[n] și screen[n] pentru a descrie poziția cursorului unde a apărut evenimentul MouseOut.

`screen[n]`	Unde `[n]` reprezintă X sau Y, folosite (împreună cu `layer[n]` și `page[n]` pentru a descrie poziția cursorului unde a apărut evenimentul `MouseOut`.

■ **`onMouseOver`:** Precizează codului scris în JavaScript că trebuie să fie executat când mouse-ul intră în zona specificată în eticheta `<AREA>`.

Proprietăți noi în JavaScript 1.2

`type`	Indică un eveniment `MouseOver`.
`target`	Indică obiectul la care a fost trimis evenimentul.
`layer[n]`	Unde `[n]` reprezintă X sau Y, folosite (împreună cu `page[n]` și `screen[n]` pentru a descrie poziția cursorului unde a apărut evenimentul `MouseOver`.
`page[n]`	Unde `[n]` reprezintă X sau Y, folosite (împreună cu `layer[n]` și `screen[n]` pentru a descrie poziția cursorului unde a apărut evenimentul `MouseOver`.
`screen[n]`	Unde `[n]` reprezintă X sau Y, folosite (împreună cu `layer[n]` și `page[n]` pentru a descrie poziția cursorului unde a apărut evenimentul `MouseOver`.

Obiectul `array` [C | 3 | I]

Obiectul `array` furnizează un mecanism de creare a tablourilor și de folosire a lor. Tablourile noi sunt create cu *`numeTablou`*`= new Array ()` sau *`numeTablou`*`= new Array (lungimeTablou)`.

Proprietăți

■ **`length`:** Un număr întreg, care reflectă numărul de elemente dintr-un tablou.
■ **`prototype`:** Este folosit pentru a adăuga proprietăți unui obiect `array`.

Metode

■ **`concat (numeTablou)`:** Concatenează elementele din două tablouri și returnează un al treilea, pe un singur nivel, fără să modifice vreunul dintre tablourile din care a derivat. (Doar în Netscape Navigator 4.0).
■ **`join (sir)`:** Returnează un șir care conține fiecare element al tabloului, separat prin șir. (Nu I).
■ **`reverse()`:** Inversează ordinea unui tablou. (Nu I).
■ **`slice(numeTablou, inceput, sfarsit)`:** Extrage o porțiune dintr-un tablou și obține un nou tablou. Parametrii `inceput` și `sfarsit` precizează elementele țintă de la care să înceapă și la care să se termine porțiunea. (Doar în Netscape Navigator 4.0.)
■ **`sort(functie)`:** Sortează un tablou conform unei funcții care definește ordinea de sortare. Funcția poate fi omisă, caz în care sortarea prestabilită se va face în ordine alfabetică. Observație: acum *`sort`* funcționează pe toate platformele.

Obiectul `button` [C | 2 | 3 | I]

Obiectul `button` definește în JavaScript un buton de apăsare dintr-un formular HTML.

Proprietăți

■ **`enabled`:** O valoare booleeană care indică dacă butonul este activat. (Nu 2 I 3).
■ **`form`:** O referire la obiectul *`form`* (formular) care conține butonul. (Nu 2 I 3).
■ **`name`:** O valoare șir care conține numele elementului buton.
■ **`type`:** O valoare șir care reflectă atributul `TYPE` al etichetei `<INPUT>`. (Nu 2 II).
■ **`value`:** O valoare șir care conține valoarea elementului buton.

Metode

■ **`click ()`:** Emulează acțiunea de executare a unui clic pe buton.
■ **`focus ()`:** Atribuie focusul butonului. (Nu 2 I 3).

Elementele de tratare a evenimentelor

■ **`onMouseDown`:** Precizează codului scris în JavaScript că trebuie să fie executat când un utilizator apasă un buton al mouse-ului.
■ **`onMouseUp`:** Precizează codului scris în JavaScript că trebuie să fie executat când un utilizator eliberează un buton al mouse-ului.
■ **`onClick`:** Precizează codului scris în JavaScript că trebuie să fie executat când se efectuează clic pe buton.
■ **`onFocus`:** Precizează codului scris în JavaScript că trebuie să fie executat când butonului îi este atribuit focusul. (Nu 2 I 3).

Obiectul `checkbox` [C | 2 | 3 | I]

Obiectul `checkbox` creează o casetă de validare într-un formular HTML disponibil în JavaScript,.

Proprietăți

■ **`checked`:** O valoare booleeană care arată dacă elementul casetă de validare este marcat.
■ **`defaultChecked`:** O valoare booleeană care arată dacă elementul casetă de validare a fost implicit marcat (adică, reflectă atributul `CHECKED`).
■ **`enabled`:** O valoare booleeană care arată dacă este activată caseta de validare. (Nu 2 I 3).
■ **`form`:** O referire la obiectul `form` (formular) care conține caseta de validare. (Nu 2 I 3).
■ **`name`:** O valoare șir care conține numele elementului casetă de validare.
■ **`type`:** O valoare șir care reflectă atributul `TYPE` din eticheta `<INPUT>`. (Nu 2 II).
■ **`value`:** O valoare șir care conține valoarea elementului casetă de validare.

Metode

■ **`click()`:** Emulează acțiunea de efectuare a unui clic pe caseta de validare.
■ **`focus()`:** Atribuie focusul casetei de validare. (Nu 2 I 3).

Elemente de tratare a evenimentelor

■ **`onClick`:** Precizează codului scris în JavaScript că trebuie să fie executat când se execută clic pe caseta de validare.
■ **`onFocus`:** Precizează codului scris în JavaScript că trebuie să fie executat când se atribuie focusul casetei de validare. (Nu 2 I 3).

Obiectul `combo` [C | I]

Obiectul `combo` definește un câmp combinat în JavaScript.

Proprietăți

■ **`enabled`:** O valoare booleeană care arată când este activată caseta combo. (Nu 2 I 3).
■ **`form`:** Reprezintă o referire către obiectul `form` care conține caseta combo. (Nu 2 I 3).
■ **`listCount`:** Un număr întreg care reflectă numărul de elemente din listă.
■ **`listIndex`:** Un număr întreg care reflectă indexul elementului selectat în listă.
■ **`multiSelect`:** O valoare booleeană care arată când câmpul combo se află în modul multiselectare.

- **name:** O valoare șir care indică numele câmpului combo.
- **value:** Reprezintă un șir care conține valoarea câmpului combo.

Metode

- **addItem(*index*):** Adaugă un articol câmpului combo înaintea articolului din poziția index.
- **click():** Simulează efectuarea unui clic pe câmpul combo.
- **clear():** Șterge conținutul câmpului combo.
- **focus():** Atribuie focusul câmpului combo.
- **removeItem(*index*):** Extrage articolul din poziția *index* din câmpul combo.

Elemente de tratare a evenimentelor

- **onClick:** Precizează codului scris în JavaScript că trebuie să fie executat când se efectuează clic cu mouse-ul pe câmpul combo.
- **onFocus:** Precizează codului scris în JavaScript că trebuie să fie executat când câmpului combo îi este atribuit focusul. (Nu 2|3).

Obiectul date [C|2|3|I]

Obiectul date furnizează proceduri pentru lucrul cu ora și data în JavaScript. Pot fi create instanțe ale obiectului folosind următoarea sintaxă.

```
NumeObiectNou = new Date(dateInfo).
```

Aici, *dateInfo* reprezintă o precizare opțională a unei anumite date și poate fi sub una dintre următoarele forme.

```
"luna ziua, anul ore:minute:secunde"
anul, luna, ziua
anul, luna, ziua, ore, minute, secunde
```

Ultimele două opțiuni reprezintă valori întregi.

Dacă nu se specifică elementul *dateInfo*, noul obiect reprezintă data și ora curente.

Proprietăți

- **prototype:** Furnizează un mecanism de adăugare a unor proprietăți unui obiect date. (Nu 2).

Metode

- **getDate():** Returnează ziua din lună pentru obiectul date curent sub forma unui număr întreg de la 1 la 31.
- **getDay():** Returnează ziua din saptămână pentru obiectul date curent, sub forma unui număr întreg de la 0 la 6 (0 reprezintă duminică, 1 reprezintă luni etc.)
- **getHours():** Returnează ora sub forma unui număr întreg de la 0 la 23 din ora obiectului date curent.
- **getMinutes():** Returnează minutele sub forma unui număr întreg de la 0 la 59 din ora obiectului date curent.
- **getMonth():** Returnează luna sub forma unui număr întreg de la 0 la 11 (0 reprezintă luna ianuarie, 1 reprezintă luna februarie etc.) pentru obiectul date curent.
- **getSeconds():** Returnează secundele sub forma unui număr întreg de la 0 la 59 pentru obiectul date curent.
- **getTime():** Returnează timpul din obiectul date curent, sub forma unui număr întreg ce reprezintă numărul de milisecunde scurse de la 1 ianuarie 1970, ora 00:00:00.

- **getTimezoneOffset():** Returnează diferența dintre ora locală și GMT, sub forma unui număr întreg, ce reprezintă numărul de minute.
- **getYear():** Returnează anul pentru obiectul date curent, sub forma unui număr întreg pe doi digiți, reprezentând anul, minus 1900.
- **parse(dateString):** Returnează numărul de milisecunde scurse între 1 ianuarie 1970, ora 00:00:00 și data specificată prin atributul *dateString*, care ar trebui să aibă următorul format: (Nu I).

```
Zi, DD Mon YYYY HH:MM:SS TZN
Luna, DD, YYYY
```

- **setDate(*dateVal*):** Stabilește ziua din lună, pentru obiectul date curent. *dateVal* este un număr întreg de la 1 la 31.
- **setHours(*oraVal*):** Stabilește ora pentru obiectul date curent. *oraVal* este un număr întreg de la 0 la 23.
- **setMinutes (*minuteVal*):** Stabilește minutele pentru obiectul date curent. *minuteVal* este un număr întreg de la 0 la 59.
- **setMonth(*lunaVal*):** Stabilește luna pentru obiectul date curent. *lunaVal* este un număr întreg de la 0 la 11 (0 reprezintă luna ianuarie, 1 reprezintă luna februarie etc.).
- **setSeconds(*secundeVal*):** Stabilește secundele pentru obiectul date curent. *secundeVal* este un număr întreg de la 0 la 59.
- **setTime(*timpVal*):** Stabilește valoarea pentru obiectul date curent. *timpVal* este un număr întreg reprezentând numărul de milisecunde scurse de la 1 ianuarie 1970, ora 00:00:00.
- **setYear(*anVal*):** Stabilește anul pentru obiectul date curent. *anVal* este un număr întreg mai mare de 1900.
- **toGMTString():** Returnează valoarea obiectului date curent, în ora GMT, sub formă de șir, folosind convențiile Internetului, astfel:

```
Zi, DD Mon YYYY HH : MM : SS GMT
```

- **toLocaleString():** Returnează valoarea obiectului date curent, în ora locală, folosind convențiile locale.
- **UTC(anVal, lunaVal, dateVal, oraVal, minuteVal, secundeVal):** Returnează numărul de milisecunde scurse de la 1 ianuarie 1970, ora 00:00:00 GMT. *anVal* este un număr întreg mai mare de 1900. *lunaVal* este un număr întreg de la 0 la 11. *dateVal* este un număr întreg de la 1 la 31. *oraVal* este un număr întreg de la 0 la 23. *minuteVal* și *secundeVal* sunt numere întregi de la 0 la 59. *oraVal*, *minuteVal* și *secundeVal* sunt opționale. (Nu I).

Obiectul document [C|2|3|I]

Obiectul document reprezintă în JavaScript atributele unui document HTML.

Proprietăți

- **alinkColor:** Definește culoarea legăturilor active, reprezentată sub forma unui șir sau triplet hexazecimal.
- **anchors:** Definește un șir de obiecte anchor, reprezentate în ordinea în care apar în documentul HTML. Folosiți anchors.length pentru a obține numărul de ancore dintr-un document.
- **applets:** Definește un tablou de obiecte miniaplicatie, reprezentate în ordinea în care apar în documentul HTML. Folosiți applets.length pentru a obține numărul de miniaplicații dintr-un document. (Nu 2).

- **bgcolor:** Definește culoarea fundalului documentului.
- **cookie:** Definește o valoare șir care conține valorile „cookie" pentru documentul curent.
- **embeds:** Definește un tablou de obiecte `plugin`, reprezentate în ordinea în care apar în documentul HTML. Folosiți `embeds.length` pentru a obține numărul de obiecte plugin dintr-un document. (Nu 2|I).
- **fgColor:** Definește culoarea primplanului documentului.
- **forms:** Definește un tablou de obiecte formular, reprezentate în ordinea în care formularele apar în fișierul HTML. Folosiți `forms.length` pentru a obține numărul de formulare dintr-un document.
- **images:** Definește un tablou de obiecte imagine, reprezentate în ordinea în care ele apar în documentul HTML. Folosiți `images.length` pentru a obține numărul de imagini dintr-un document. (Nu 2|I).
- **lastModified:** Definește o valoare șir care conține data ultimei modificări a documentului.
- **linkColor:** Definește culoarea legăturilor, furnizată sub forma unui șir sau a unui triplet hexazecimal.
- **links:** Definește un șir de obiecte de legătură, date în ordinea în care legăturile hipertext apar în documentul HTML. Folosiți `links.length` pentru a obține numărul de legături dintr-un document.
- **location:** Definește un șir care conține adresa URL a documentului curent. Folosiți `document.URL` în loc de `document.location`. Această proprietate va fi exclusă la următoarea versiune.
- **referrer:** Reprezintă o valoare șir care conține adresa URL a documentului apelant, când utilizatorul urmărește o legătură.
- **title:** Definește un șir care conține titlul documentului curent.
- **URL:** Definește un șir care reprezintă adresa URL a documentului curent. Se folosește în loc de `document.location` (Nu I).
- **vlinkColor:** Definește culoarea legăturilor urmate, reprezentată ca un șir sau ca un triplet hexazecimal.

Elemente de tratare a evenimentelor

- **onMouseDown:** Precizează codului scris în JavaScript că trebuie să fie executat când utilizatorul execută clic pe un buton al mouse-ului.
- **onMouseUp:** Precizează codului scris în JavaScript că trebuie să fie executat când utilizatorul eliberează un buton al mouse-ului.
- **onKeyUp:** Precizează codului scris în JavaScript că trebuie să fie executat când utilizatorul eliberează o anumită tastă. (Doar în Netscape Navigator 4.0) (4)
- **onKeyPress:** Precizează codului scris în JavaScript că trebuie să fie executat când utilizatorul ține apăsată o anumită tastă. (Doar în Netscape Navigator 4.0) (4)
- **onKeyDown:** Precizează codului scris în JavaScript că trebuie să fie executat când utilizatorul apasă pe o anumită tastă. (Doar în Netscape Navigator 4.0) (4)
- **onDblClick:** Precizează codului scris în JavaScript că trebuie să fie executat când utilizatorul execută dublu clic în zona respectivă. (Nu este implementat pe Macintosh). (Doar în Netscape Navigator 4.0) (4)

Metode

- **captureEvents():** Este folosită într-o fereastră cu cadre (împreună cu `enableExternalCapture`) și precizează că fereastra va captura toate evenimentele specificate. Este un element nou în JavaScript 1.2.

- **clear():** Șterge fereastra documentului. (Nu I).
- **close():** Închide fluxul curent de ieșire.
- **open(*mimeType*):** Deschide un flux ce permite ca metodele `write()` și `writeln()` să scrie în fereastra document. *mimeType* este un șir opțional care precizează tipul unui document acceptat de Netscape Navigator sau un element plugin. (de exemplu, `text/html` sau `image/gif`).
- **releaseEvents(*eventType*):** Precizează că fereastra curentă trebuie să elibereze evenimentele (spre deosebire de capturare), astfel încât aceste evenimente să poată fi trecute altor obiecte, probabil în continuare, în ierarhia evenimentelor. Este un element nou în JavaScript 1.2.
- **routeEvent(event):** Trimite sau direcționează un eveniment trecând prin ierarhia normală a evenimentelor.
- **write():** Prin această metodă se poate scrie un text și un HTML în documentul specificat.
- **writeln():** Prin această metodă se poate scrie un text și un HTML în documentul specificat, urmat de un caracter de linie nouă.

Obiectul `fileUpload` [C | 3]

Definește un element de descărcare a unui fișier, într-un formular HTML.

Proprietăți

- **name:** Definește o valoare șir care reflectă numele elementului de descărcare fișier.
- **value:** Definește o valoare șir care reflectă câmpul elementului de descărcare fișier.

Obiectul `form` [C | 2 | 3 | I]

Obiectul `form` definește un formular HTML în JavaScript. Fiecare formular HTML dintr-un document este reprezentat de o instanțiere distinctivă a obiectului formular.

Proprietăți

- **action:** Definește o valoare șir care precizează adresa URL către care sunt trimise datele din formular.
- **elements:** Definește un tablou de obiecte pentru fiecare element formular, în ordinea în care apar în formular.
- **encoding:** Definește un șir care conține codificarea MIME a formularului, așa cum s-a precizat în atributul `ENCTYPE`.
- **method:** Definește o valoare șir care conține metoda de trimitere a datelor formularului către server.
- **target:** Definește o valoare șir care conține numele ferestrei la care sunt direcționate răspunsurile la trimiterile formularului.

Metode

- **reset():** Reinițializează formularul. (Nu 2|I).
- **submit():** Trimite formularul.

Elemente de tratare a evenimentelor

- **onReset:** Precizează codului scris în JavaScript că trebuie să fie executat când formularul este reinițializat. (Nu 2|I).
- **onSubmit:** Precizează codului scris în JavaScript că trebuie să fie executat când formularul este trimis. Codul ar trebui să returneze o valoare `true` (adevărat) pentru a permite ca formularul să fie trimis. O valoare `false` (fals) împiedică transmiterea formularului.

Obiectul frame [C | 2 | 3 | I]

Obiectul frame definește o fereastră cadru în JavaScript.

Proprietăți

- **frames:** Definește un tablou de obiecte pentru fiecare cadru dintr-o fereastră. Cadrele apar în tablou în ordinea în care apar în codul sursă HTML.
- **onblur:** Definește un șir care reflectă elementul de tratare a evenimentului onBlur, pentru cadru. Se pot atribui noi valori acestei proprietăți pentru a modifica elementul de tratare a evenimentului. (Nu 2).
- **onfocus:** Definește un șir care reflectă elementul de tratare a evenimentului onFocus pentru cadru. Se pot atribui noi valori acestei proprietăți pentru a modifica elementul de tratare a evenimentelor. (Nu 2).
- **parent:** Definește un șir care indică numele ferestrei ce conține configurația cadrului.
- **self:** Definește o alternativă pentru numele ferestrei curente.
- **top:** Definește o alternativă pentru numele ferestrei de deasupra.
- **window:** Definește o alternativă pentru numele ferestrei curente.

Metode

- **alert(mesaj):** Afișează mesaj într-o casetă de dialog.
- **blur():** Retrage focusul cadrului. (Nu 2).
- **clearInterval (intervalID):** Anulează intervalele de așteptare create de metoda setInterval. Este un element nou în JavaScript 1.2.
- **close():** Închide fereastra.
- **confirm(mesaj):** Afișează mesaj într-o casetă de dialog cu butoane de tip OK și Cancel. Returnează true (adevărat) sau false (fals) în funcție de butonul pe care utilizatorul a executat clic.
- **focus():** Atribuie focusul cadrului. (Nu 2)
- **open(url,nume,caracteristici):** Deschide o adresă URL într-o fereastră denumită cu nume. Dacă nume nu există, se creează o nouă fereastră cu acel nume. Atributul caracteristici este un argument de tip șir opțional, care conține o listă de caracteristici pentru noua fereastră. Lista de caracteristici conține oricare dintre următoarele perechi nume-valoare, separate prin virgule și fără spații libere:

toolbar=[yes, no, 1, 0]	Precizează dacă fereastra ar trebui să aibă o bară cu instrumente.
location=[yes, no, 1, 0]	Precizează dacă fereastra ar trebui să aibă un câmp de locație.
directories=[yes, no, 1, 0]	Precizează dacă fereastra ar trebui să aibă butoane pentru directoare.
status=[yes, no, 1, 0]	Precizează dacă fereastra ar trebui să aibă o bară de stare.
menubar=[yes, no, 1, 0]	Precizează dacă fereastra ar trebui să aibă meniuri.
scrollbars=[yes,no,1,0]	Precizează dacă fereastra ar trebui să aibă bare de derulare.
resizable=[yes,no,1,0]	Precizează dacă fereastra ar trebui să poată fi redimensionată.

width=pixeli	Precizează lățimea ferestrei în pixeli.
height=pixeli	Precizează înălțimea ferestrei în pixeli.

- **print():** Tipărește conținutul unei ferestre sau al unui cadru. Este echivalent cu a executa clic pe butonul Print (tipărire) din browserul Netscape Navigator. Este un element nou în Java Script 1.2.
- **prompt(*mesaj,raspuns*):** Afișează mesajul într-o casetă de dialog, care conține un câmp de introducere text, având valoarea prestabilită dată de *raspuns*. Răspunsul utilizatorului din câmpul de introducere text va fi returnat sub formă de șir.
- **setInterval(*functie, msec, [args]*):** Apelează repetat o funcție după scurgerea perioadei date în milisecunde prin msec. Este un element nou în JavaScript 1.2.
- **setInterval(*expresie, msec*):** Evaluează *expresia* după scurgerea perioadei date prin parametrul msec. Este un element nou în JavaScript 1.2.
- **setTimeout(*expresie, timp*):** Evaluează *expresia* după intervalul *timp*; unde *timp* este o valoare dată în milisecunde. Intervalul poate fi denumit folosind următoarea structură:

```
nume=setTimeOut(expresie, timp)
```

- **clearTimeout(*nume*):** Anulează intervalul de așteptare cu numele nume.

Elemente de tratare a evenimentelor

- **onBlur:** Precizează codului scris în JavaScript că trebuie să fie executat când este retras focusul dintr-un cadru. (Nu 2).
- **onFocus:** Precizează codului scris în JavaScript că trebuie să fie executat când este atribuit focusul unui cadru. (Nu 2).
- **onMove:** Precizează codului scris în JavaScript că trebuie să fie executat când utilizatorul mută un cadru. (Doar în Netscape Navigator 4.0).
- **onResize:** Precizează codului scris în JavaScript că trebuie să fie executat când utilizatorul redimensionează un cadru. (Doar în Netscape Navigator 4.0).

Obiectul function [C | 3]

Obiectul function furnizează un mecanism pentru a preciza codului JavaScript că trebuie compilat ca o funcție. Iată sintaxa de folosire a obiectului function:

```
numeFunctie = new Function(arg1, arg2, arg3, ...,codFunctie)
```

Această sintaxă este identică cu următoarea sintaxă:

```
function numeFunctie (arg1, arg2, arg3, ...) {
   codFunctie
}
```

Totuși, în prima sintaxă, *numeFunctie* este o variabilă cu o referire la funcție, iar funcția este evaluată de fiecare dată când este folosită în loc să fie compilată o dată.

Proprietăți

- **arguments:** Definește un număr întreg care reflectă numărul de argumente dintr-o funcție.
- **prototype:** Furnizează un mecanism de adăugare a proprietăților la un obiect function.

Obiectul hidden [C | 2 | 3 | I]

Obiectul hidden reflectă în JavaScript un câmp ascuns dintr-un formular HTML

Proprietăți

- **name:** Definește o valoare șir care conține numele elementului ascuns.
- **type:** Definește o valoare șir care reflectă proprietatea TYPE a etichetei <INPUT>. (Nu 2 | I).
- **value:** Definește o valoare șir care conține valoarea elementului text ascuns.

Obiectul history [C | 2 | 3 | I]

Obiectul history permite unui script să funcționeze în JavaScript împreună cu lista de istorice a browserului Netscape Navigator. Din motive de securitate și confidențialitate, conținutul listei nu este reprezentat în JavaScript.

Proprietăți

- **length:** Definește un număr întreg care reflectă numărul de articole din lista de istorice. (Nu I).

Metode

- **back():** Revine la documentul anterior din lista de istorice .(Nu I).
- **forward():** Trece la următorul document din lista de istorice. (Nu I).
- **go(locatie):** Trece la documentul din lista de istorice precizat de atributul *locatie*, care poate fi un șir sau un număr întreg. Dacă este un șir, el reflectă adresa URL, parțial sau în totalitate, în lista de istorice. Dacă este un număr întreg, *locatie* reprezintă poziția relativă a documentului în lista de istorice. Ca și un număr întreg, *locatie* poate avea valoare pozitivă sau negativă. (Nu I).

Obiectul image [C | 3]

Obiectul image reprezintă o imagine inclusă într-un document HTML.

Proprietăți

- **border:** Definește un număr întreg care reprezintă lățimea bordurii imaginii, dată în pixeli.
- **complete:** Definește o valoare booleană, care indică dacă s-a terminat încărcarea imaginii.
- **height:** Definește un număr întreg, care reprezintă înălțimea unei imagini, în pixeli.
- **hspace:** Definește un număr întreg care reprezintă atributul HSPACE al etichetei <IMG>.
- **lowsrc:** Definește un șir care conține adresa URL a versiunii de rezoluție mică a imaginii ce trebuie încărcată.
- **name:** Definește un șir care indică numele obiectului image (imagine).
- **prototype:** Furnizează un mecanism pentru adăugarea de proprietăți ca un obiect image.
- **src:** Definește o valoare șir, care indică adresa URL a imaginii.
- **vspace:** Definește un număr întreg care reprezintă atributul VSPACE al etichetei <IMG>.
- **width:** Definește un număr întreg care indică lățimea unei imagini, dată în pixeli.

Elemente de tratare a evenimentelor

- **onKeyUp:** Precizează codului scris în JavaScript că trebuie să fie executat când utilizatorul eliberează o anumită tastă. (Doar în Netscape Navigator 4.0) (4).
- **onKeyPress:** Precizează codului scris în JavaScript că trebuie să fie executat când utilizatorul ține apăsată o anumită tastă. (Doar în Netscape Navigator 4.0) (4).

- **onKeyDown:** Precizează codului scris în JavaScript că trebuie să fie executat când utilizatorul apasă o anumită tastă. (Doar în Netscape Navigator 4.0) (4)
- **onAbort:** Precizează codului scris în JavaScript că trebuie să fie executat dacă încercarea de încărcare a imaginii este abandonată. (Nu 2).
- **onError:** Precizează codului scris în JavaScript că trebuie să fie executat dacă apare o eroare în timp ce se încarcă imaginea. Stabilirea acestui element de tratare a evenimentelor la valoarea 0 elimină mesajele de eroare în cazul apariției unei erori în timpul încărcării. (Nu 2).
- **onLoad:** Precizează codului scris în JavaScript că trebuie să fie executat când imaginea este încărcată. (Nu 2).

Obiectul layer [4] (Doar în Netscape Navigator 4.0).

Obiectul layer (strat) este folosit pentru a insera conținutul structurat pe mai multe straturi într-o pagină, ele putând fi ascunse sau nu. Fiecare strat este accesibil prin cod scris în JavaScript. Cea mai uzuală aplicație a straturilor constă în dezvoltarea paginilor Dynamic HTML (DHTML). Cu ajutorul straturilor puteți crea un conținut animat sau cu alte efecte dinamice într-o pagină, prin trecerea ciclică prin straturile pe care le-ați definit.

Proprietăți

- **above:** Plasează un strat deasupra unui strat nou creat.
- **backgroud:** Este folosit pentru a reprezenta stratul din fundal sub forma unor imagini mozaicate.
- **below:** Plasează un strat sub un strat nou creat.
- **bgColor:** Stabilește culoarea din fundal a stratului.
- **clip(left, top, right, bottom):** Precizează marginile vizibile ale stratului.
- **height:** Precizează înălțimea stratului, exprimată în pixeli (număr întreg) sau într-un procent din stratul curent.
- **ID:** A fost numită anterior NAME. Este folosită pentru a denumi stratul, astfel acesta poate a fi apelat după nume și accesat de alt cod JavaScript.
- **left:** Precizează poziționarea pe orizontală a colțului din stânga sus al stratului. Este folosit cu proprietatea TOP.
- **page[n]:** Unde [n] este X sau Y. Precizează poziționarea pe orizontală (X) sau verticală (Y) a colțului din stânga-sus al stratului, în funcție de restul documentului ce-l conține.

(Observație: Această proprietate este diferită de proprietățile Left și Top).

- **parentLayer:** Precizează obiectul layer (strat) care conține stratul actual.
- **SRC:** Precizează sursa HTML ce trebuie să fie afișată o dată cu stratul țintă. (Această sursă poate include de asemenea cod JavaScript).
- **siblingAbove:** Precizează obiectul layer aflat imediat deasupra stratului actual.
- **siblingBelow:** Precizează obiectul layer aflat imediat sub stratul prezent.
- **top:** Precizează poziționarea pe verticală a colțului din stânga sus a stratului. (Este folosit împreună cu proprietatea Left).
- **visibility:** Precizează nivelul de vizibilitate al stratului. Exist[trei variante: show (este vizibil), hidden (nu este vizibil) și inherit (stratul moștenește proprietățile părintelui său).
- **width:** Precizează lățimea stratului. Este folosit pentru proceduri de aranjare automată a textului. Astfel lățimea reprezintă marginea după care conținutul se aranjează automat în interiorul stratului.

■ **z-index:** Precizează poziția z (sau ordinea în stivă) a stratului. Este folosit pentru a stabili poziția stratului, relativ la ordinea generală de rotație a tuturor straturilor. Este exprimat ca și un număr întreg. (Este folosit acolo unde există multe straturi).

Evenimente

■ **onBlur:** Precizează codului scris în JavaScript că trebuie să fie executat când stratul pierde focusul.
■ **onFocus:** Precizează codului scris în JavaScript că trebuie să fie executat când stratului îi este atribuit focusul.
■ **onLoad:** Precizează codului scris în JavaScript că trebuie să fie executat când este încărcat un strat.
■ **onMouseOut:** Precizează codului scris în JavaScript că trebuie să fie executat când cursorul mouse-ului se mută în afara stratului.

Proprietăți noi

`type`	Indică un eveniment `MouseOut`.
`target`	Indică obiectul la care a fost trimis evenimentul.
`layer[n]`	Unde `[n]` reprezintă X sau Y folosite (împreună cu `page[n]` și `screen[n]`) pentru a descrie poziția cursorului unde a apărut evenimentul `MouseOut`.
`page[n]`	Unde `[n]` reprezintă X sau Y folosite (împreună cu `layer[n]`) și `screen[n]`) pentru a descrie poziția cursorului unde a apărut evenimentul `MouseOut`.
`screen[n]`	Unde `[n]` reprezintă X sau Y folosite (împreună cu `layer[n]` și `page[n]`) pentru a descrie poziția cursorului unde a apărut evenimentul `MouseOut`.

■ **onMouseOver:** Precizează codului scris în JavaScript că trebuie să fie executat când cursorul mouse-ului intră pe strat.

Proprietăți noi în JavaScript 1.2

`type`	Indică un eveniment `MouseOver`.
`target`	Indică obiectul la care a fost trimis evenimentul.
`layer[n]`	Unde `[n]` reprezintă X sau Y folosite (împreună cu `page[n]` și `screen[n]`) pentru a descrie poziția cursorului unde a apărut evenimentul `MouseOver`.
`page[n]`	Unde `[n]` reprezintă X sau Y folosite (împreună cu `layer[n]` și `screen[n]`) pentru a descrie poziția cursorului unde a apărut evenimentul `MouseOver`.
`screen[n]`	Unde `[n]`reprezintă X sau Y folosite (împreună cu `layer[n]` și `page[n]`) pentru a descrie poziția cursorului unde a apărut evenimentul `MouseOver`.

Metode

■ **captureEvents():** Este folosită într-o fereastră cu cadre (împreună cu `enableExternalCapture`) și arată că fereastra va captura toate evenimentele specificate. Este un element nou în JavaScript 1.2.
■ **load(sursa, latime):** Modifică sursa stratului prin înlocuirea ei cu un HTML (sau JavaScript) din fișierul specificat în *sursa*. Folosind această metodă, puteți de asemenea transmite o valoare de lățime (în pixeli) pentru a încadra noul conținut.
■ **moveAbove(strat):** Plasează stratul deasupra stratului indicat din stivă.
■ **moveBelow(strat):** Plasează stratul sub stratul indicat din stivă.

■ **moveBy(x,y):** Modifică poziția stratului conform valorilor precizate, exprimate în pixeli.
■ **moveTo(x,y):** Modifică poziția stratului (în interiorul stratului care îl conține) la poziția dată de coordonatele specificate, exprimate în pixeli.
■ **moveToAbsolute(x,y):** Modifică poziția stratului (în interiorul paginii) la poziția dată de coordonatele specificate, exprimate în pixeli.
■ **releaseEvents(eventType):** Precizează ca fereastra curentă ar trebui să elibereze evenimentele și nu să le captureze, astfel că aceste evenimente pot fi atribuite altor obiecte, probabil în continuare în ierarhia evenimentelor. Este un element nou în JavaScript 1.2.
■ **resizeBy(latime, inaltime):** Redimensionează stratul cu valorile specificate, exprimate în pixeli.
■ **resizeTo(latime, inaltime):** Redimensionează stratul la lățimea și înălțimea specificate, exprimate în pixeli.
■ **routeEvent(event):** Trimite sau direcționează un eveniment folosind ierarhia normală a evenimentelor.

Obiectul `link` [C | 2 | 3 | I]

Obiectul `link` reprezintă o legătură hipertext în corpul unui document.

Proprietăți

■ **hash:** Definește o valoare șir care conține numele ancorei din adresa URL.
■ **host:** Definește o valoare șir care conține numele gazdei și numărul portului din adresa URL.
■ **hostname:** Definește o valoare șir care conține numele domeniului (sau adresa IP numerică) din adresa URL.
■ **href:** Definește o valoare șir care conține întreaga adresă URL.
■ **pathname:** Definește o valoare șir care specifică porțiunea cu calea din adresa URL.
■ **port:** Definește o valoare șir care conține numărul portului din adresa URL.
■ **protocol:** Definește o valoare șir care conține protocolul din adresa URL (inclusiv „:", dar nu și barele de separare).
■ **search:** Definește o valoare șir care conține toate informațiile trecute unui apel `GET` CGI-BIN (cum ar fi toate informațiile trecute după semnul de întrebare).
■ **target:** Definește o valoare șir care conține numele ferestrei sau cadrului specificat în atributul `TARGET`.

Elemente de tratare a evenimentelor

■ **onMouseDown:** Precizează codului scris în JavaScript că trebuie să fie executat când un utilizator apasă un buton al mouse-ului. (Doar în JavaScript 1.2 și Netscape Navigator 4.0) (4).
■ **onMouseOut:** Precizează codului scris în JavaScript că trebuie să fie executat când utilizatorul mută cursorul mouse-ului în afara unui obiect. (Doar în JavaScript 1.2 și Netscape Navigator 4.0) (4).

Proprietăți noi în JavaScript 1.2

`type`	Indică un eveniment `MouseOut`.
`target`	Indică obiectul la care a fost trimis evenimentul.
`layer[n]`	Unde `[n]` reprezintă X sau Y folosite (împreună cu `page[n]` și `screen[n]`) pentru a descrie poziția cursorului unde a apărut evenimentul `MouseOut`.
`page[n]`	Unde `[n]` reprezintă X sau Y folosite (împreună cu `layer[n]` și `screen[n]`) pentru a descrie poziția cursorului unde a apărut evenimentul `MouseOut`.

`screen[n]`	Unde `[n]` reprezintă X sau Y folosite (împreună cu `layer[n]` și `page[n]`) pentru a descrie poziția cursorului unde a apărut evenimentul `MouseOut`.

- **`onMouseUp`:** Precizează codului scris în JavaScript că trebuie să fie executat când utilizatorul eliberează un buton al mouse-ului.
- **`onKeyUp`:** Precizează codului scris în JavaScript că trebuie să fie executat când utilizatorul eliberează o anumită tastă. (Doar în Netscape Navigator 4.0) (4)
- **`onKeyPress`:** Precizează codului scris în JavaScript că trebuie să fie executat când utilizatorul ține apăsată o anumită tastă. (Doar în Netscape Navigator 4.0) (4)
- **`onKeyDown`:** Precizează codului scris în JavaScript că trebuie să fie executat când utilizatorul apasă o anumită tastă. (Doar în Netscape Navigator 4.0) (4).
- **`onDblClick`:** Precizează codului scris în JavaScript că trebuie să fie executat când utilizatorul execută dublu clic în zona respectivă. (Nu este implementat pe Macintosh). (Doar în Netscape Navigator 4.0) (4)
- **`moveMouse`:** Precizează codului scris în JavaScript că trebuie să fie executat când pointerul mouse-ului se mută pe legătură. (Nu 2|3).
- **`onClick`:** Precizează codului scris în JavaScript că trebuie să fie executat când se execută clic pe legătură.
- **`onMouseOver`:** Precizează codului scris în JavaScript că trebuie să fie executat când pointerul mouse-ului se mută pe legătura hipertext.

Proprietăți noi în JavaScript 1.2

`type`	Indică un eveniment `MouseOver`.
`target`	Indică obiectul la care a fost trimis evenimentul.
`layer[n]`	Unde `[n]` reprezintă X sau Y folosite (împreună cu `page[n]` și `screen[n]`) pentru a descrie poziția cursorului unde a apărut evenimentul `MouseOver`.
`page[n]`	Unde `[n]` reprezintă X sau Y folosite (împreună cu `layer[n]` și `screen[n]`) pentru a descrie poziția cursorului unde a apărut evenimentul `MouseOver`.
`screen[n]`	Unde `[n]` reprezintă X sau Y folosite (împreună cu `layer[n]` și `page[n]`) pentru a descrie poziția cursorului unde a apărut evenimentul `MouseOver`.

Obiectul `location` [C | 2 | 3 | I]

Obiectul `location` reprezintă informația referitoare la adresa URL curentă.

Proprietăți

- **`hash`:** Definește o valoare șir care conține numele ancorei din adresa URL.
- **`host`:** Definește o valoare șir care conține numele gazdei și numărul portului din adresa URL.
- **`hostname`:** Definește o valoare șir care conține numele domeniului (sau adresa IP numerică) din adresa URL.
- **`href`:** Definește o valoare șir care conține întreaga adresă URL.
- **`pathname`:** Definește o valoare șir care precizează porțiunea cu calea din adresa URL.
- **`port`:** Definește o valoare șir care conține numărul portului din adresa URL.
- **`protocol`:** Definește o valoare șir care conține protocolul din adresa URL (inclusiv „:", dar nu și barele de separare).
- **`search`:** Reprezintă o valoare șir care conține toate informațiile trecute unui apel `GET` CGI-BIN (cum ar fi toate informațiile trecute după semnul de întrebare).

Metode

- **`reload()`:** Reîncarcă documentul curent. (Nu 2|I).
- **`replace(url)`:** Încarcă adresa *`url`* peste intrarea curentă din lista de istorice, astfel va fi imposibilă navigarea înapoi la adresa URL anterioară folosind butonul Back. (Nu 2|I).

Obiectul `math` [C | 2 | 3 | I]

Obiectul `math` furnizează proprietăți și metode pentru calcule matematice avansate.

Proprietăți

- **`E`:** Reprezintă valoarea constantei lui Euler (rotunjită la 2.718), folosită ca bază a logaritmilor naturali.
- **`LN10`:** Reprezintă valoarea logaritmului natural al lui 10 (rotunjită la 2.302).
- **`LN2`:** Reprezintă valoarea logaritmului natural al lui 2 (rotunjită la 0.693).
- **`LOG10E`:** Reprezintă valoarea logaritmului în baza 10 al lui e (rotunjită la 0.434).
- **`LOG2E`:** Reprezintă valoarea logaritmului în baza 2 al lui e (rotunjită la 1.442).
- **`PI`:** Reprezintă valoarea numărului pi, folosită pentru a calcula circumferința și aria cercurilor (rotunjită la 3.1415).
- **`SQRT1_2`:** Reprezintă valoarea rădăcinii pătrate din 1/2 (rotunjită la 0.707)
- **`SQRT2`:** Reprezintă valoarea rădăcinii pătrate din 2 (rotunjită la 1.414)

Metode

- **`abs(numar)`:** Returnează valoarea absolută a lui *`numar`*. Valoarea absolută reprezintă valoarea unui număr pentru care semnul este ignorat, astfel `abs(4)` și `abs(-4)` vor returna ambele valoarea `4`.
- **`acos(numar)`:** Returnează arc-cosinusul lui *`numar`* în radiani.
- **`asin(numar)`:** Returnează arc-sinusul lui *`numar`* în radiani.
- **`atan(numar)`:** Returnează arc-tangenta lui *`numar`* în radiani.
- **`atan2(numar1, numar2)`:** Returnează unghiul coordonatelor polare corespunzătoare coordonatelor carteziene (`numar1`, `numar2`). (Nu I).
- **`ceil(numar)`:** Returnează următorul număr întreg mai mare decât *`numar`*; cu alte cuvinte , rotunjește la următorul număr întreg.
- **`cos(numar)`:** Returnează cosinusul lui *`numar`*, care reprezintă un unghi dat în radiani.
- **`exp(numar)`:** Returnează valoarea lui `E` la puterea *`numar`*.
- **`floor(numar)`:** Returnează următorul număr întreg mai mic decât *`numar`*, cu alte cuvinte rotunjește la numărul întreg anterior.
- **`log(numar)`:** Returnează logaritmul natural al lui *`numar`*.
- **`max(numar1, numar 2)`:** Returnează cel mai mare număr dintre *`numar1`* și *`numar2`*.
- **`min(numar1, numar 2)`:** Returnează cel mai mic număr dintre *`numar1`* și *`numar2`*.
- **`pow(numar1, numar 2)`:** Returnează *`numar1`* la puterea *`numar2`*.
- **`random()`:** Returnează un număr aleator între 0 și 1 (în momentul lansării, această metodă era disponibilă numai în versiunea pe UNIX a browserului Navigator 2.0).
- **`round(numar)`:** Rotunjește *`numar`* la cel mai apropiat număr întreg.
- **`sin(numar)`:** Returnează sinusul lui *`numar`*, care reprezintă un unghi dat în radiani.
- **`sqrt(numar)`:** Returnează rădăcina pătrată din *`numar`*.
- **`tan(numar)`:** Returnează tangenta de *`numar`*, care reprezintă un unghi în radiani.

Obiectul mimetype [C | 3]

Obiectul mimetype reprezintă un tip MIME acceptat de browserul client.

Proprietăți

- **type:** Definește o valoare șir reprezentând tipul MIME.
- **description:** Definește un șir care conține o descriere a tipului MIME.
- **enabledPlugin:** Definește o referire la obiectul plugin pentru programul plug-in care acceptă tipul MIME.
- **suffixes:** Definește un șir care conține o listă de extensii de fișiere separate prin virgule, pentru tipul MIME.

Obiectul navigator [C | 2 | 3 | I]

Obiectul navigator prezintă informații despre versiunea browserului Navigator folosită.

Proprietăți

- **appCodeName:** Definește o valoare șir care conține numele codului client (de exemplu „Mozilla" pentru Netscape Navigator).
- **appName:** Definește o valoare șir care conține numele clientului (de exemplu „Netscape" pentru Netscape Navigator).
- **appVersion:** Definește o valoare șir care conține informația despre versiune pentru client, în următoarea formă:

```
numarVersiune (platforma; tara)
```

De exemplu, Netscape Navigator 2.0 versiunea beta 6 pentru Windows 95 (versiunea internațională), va avea proprietatea appVersion cu valoarea `2.0b6 (Win32; I)`.

- **language:** Precizează traducerea browserului Navigator. (Este o proprietate protejată la scriere). Este un element nou în JavaScript 1.2.
- **mimeTypes:** Este un tablou de obiecte mimeTypes reprezentând tipurile MIME acceptate de browserul client. (Nu 2 | I).
- **platform:** Precizează platforma pentru care a fost compilat browserul Navigator. (De exemplu, Win32, MacPPC, UNIX.) Este un element nou în JavaScript 1.2.
- **plugins:** Este un tablou de obiecte plugin reprezentând programele plug-in dintr-un document în ordinea apariției lor în documentul HTML. (Nu 2 | I).
- **userAgent:** Este un șir care conține valoarea completă a antetului de agent pentru utilizator trimis în cererea HTTP. Următorul șir conține toate informațiile din appCodeName și appVersion:

```
Mozilla/2.0b6  (Win32; I)
```

Metode

- **javaEnabled():** Returnează o valoare booleeană care precizează dacă limbajul Java este activat în browser. (Nu 2 | I).
- **preference(preference.Name, setValue):** La scripturile semnate, această metodă permite programatorului să stabilească anumite preferințe referitoare la browser. Preferințele disponibile folosind această metodă sunt:

general.always_load_images	O valoare true/false (adevărat/fals) care stabilește încărcarea automată sau nu a imaginilor
security.enable_java	O valoare true/false care stabilește dacă Java este activat sau nu.

javascript.enabled	O valoare true/false care stabilește dacă JavaScript este activat sau nu.
browser.enable_style_sheets	O valoare true/false care stabilește dacă foile de stil sunt activate sau nu.
autoupdate.enabled	O valoare true/false care stabilește dacă autoinstall este activat sau nu.
network.cookie.cookieBehaviour	(0,1,2) Reprezintă o valoare care stabilește modul în care sunt tratate elementele cookie. Sunt trei parametri 0 – acceptă toate elementele cookie, 1 – acceptă doar acele elemente cookie care sunt trimise serverului de origine, 2 refuză orice element cookie.
network.cookie.warnAboutCookies	O valoare true/false care stabilește dacă browserul vă va atenționa în cazul acceptării elementelor cookie.

Obiectul option [C | 3]

Obiectul option este folosit pentru a crea intrări într-o listă de selecții, folosind următoarea sintaxă:

```
optionName = new Option(optionText, optionValue, defaultSelected, selected)
```

Apoi este folosită următoarea linie:

```
selectName.options[index] = optionName.
```

Proprietăți

- **defaultSelected:** Definește o valoare booleeană care precizează dacă opțiunea este selectată în mod prestabilit.
- **index:** Definește o valoare întreagă care precizează indexul opțiunii, din lista de selecție.
- **prototype:** Furnizează un mecanism de adăugare a proprietăților la un obiect option.
- **selected:** Definește o valoare booleeană care indică dacă opțiunea este selectată curent.
- **text:** Definește o valoare șir care reprezintă textul afișat pentru opțiune.
- **value:** Definește o valoare șir care indică valoarea trimisă serverului când se trimite formularul.

Obiectul password [C | 2 | 3 | I]

Obiectul password reprezintă un câmp text pentru parolă dintr-un formular HTML în JavaScript.

Proprietăți

- **defaultValue:** Este o valoare șir care conține valoarea prestabilită a elementului parolă (cum ar fi valoarea atributului VALUE).
- **enabled:** Este o valoare booleeană care precizează dacă este activat câmpul parolă. (Nu 2 | 3).
- **form:** Reprezintă o referință către obiectul form (formular) care conține câmpul parolă. (Nu 2 | 3).
- **name:** Reprezintă o valoare șir care conține numele elementului parolă.
- **value:** Reprezintă o valoare șir care conține valoarea elementului parolă.

Metode

- **focus():** Emulează acțiunea de acordare a focusului în câmpul parolă.
- **blur():** Emulează acțiunea de retragere a focusului din câmpul parolă.
- **select():** Emulează acțiunea de selectare a textului în câmpul parolă.

Elemente de tratare a evenimentelor

- **onBlur:** Precizează codului scris în JavaScript că trebuie să fie executat când câmpul parolă pierde focusul. (Nu 2|3).
- **onFocus:** Precizează codului scris în JavaScript că trebuie să fie executat când câmpul parolă primește focusul. (Nu 2|3).

Obiectul plugin

Obiectul plugin reprezintă un program plug-in acceptat de către browser.

Proprietăți

- **name:** Definește o valoare șir care reprezintă numele programului plug-in.
- **filename:** Definește o valoare șir care reprezintă numele de fișier al programului plug-in pe discul sistemului.
- **description:** Definește o valoare șir care conține descrierea furnizată de programul plug-in.

Obiectul radio [C|2|3|I]

Obiectul radio reprezintă un set de butoane radio dintr-un formular HTML în JavaScript. Pentru a accesa butoane radio individuale, folosiți indexuri numerice începând de la zero. De exemplu, butoanele individuale dintr-un set de butoane radio numit testRadio, pot fi referite de testRadio[0], testRadio[1] etc.

Proprietăți

- **checked:** Definește o valoare booleeană, care precizează dacă un anumit buton radio este marcat. Poate fi folosită pentru a selecta sau deselecta un buton.
- **defaultChecked:** Este o valoare booleeană care precizează dacă un anumit buton radio a fost marcat în mod prestabilit (dacă da, el reflectă atributul CHECKED). (Nu I).
- **enabled:** Definește o valoare booleeană care precizează dacă butonul radio este activat. (Nu 2|3).
- **form:** Definește o referință la obiectul form care conține butonul radio. (Nu 2|3).
- **length:** Definește o valoare întreagă care precizează numărul de butoane radio din set. (Nu I).
- **name:** Definește o valoare șir care conține numele setului de butoane radio.
- **value:** Definește o valoare șir care conține valoarea unui anumit buton radio dintr-un set (astfel, aceasta reflectă atributul VALUE).

Metode

- **click():** Emulează acțiunea de executare a unui clic pe butonul radio.
- **focus():** Atribuie focusul butonului radio. (Nu 2|3).

Elemente de tratare a evenimentelor

- **onClick:** Precizează codului scris în JavaScript că trebuie să fie executat când se execută clic pe un buton radio.
- **onFocus:** Precizează codului scris în JavaScript că trebuie să fie executat când un buton radio primește focusul. (Nu 2|3).

Obiectul regExp

Obiectul regExp este relevant la acțiunea de căutare a expresiilor regulate. Proprietățile lui sunt stabilite înainte sau după ce se realizează o căutare. Ele, în general, nu dețin controlul asupra operației de căutare în sine, dar articulează o serie de valori care pot fi accesate prin căutare.

Proprietăți

- **input:** Reprezintă șirul după care este comparată o expresie regulată. Este un element nou în JavaScript 1.2.
- **multiline [true, false]:** Stabilește dacă se continuă căutarea și dincolo de întreruperile de linie în liniile multiple (true – adevărat) sau nu (false – fals). Este un element nou în JavaScript 1.2.
- **lastMatch:** Este o proprietate care precizează ultimele caractere care s-au potrivit. Este un element nou în JavaScript 1.2.
- **lastParen:** Este o proprietate care precizează ultimul șir potrivit care apare în paranteze. Este un element nou în JavaScript 1.2.
- **leftContext:** Este o proprietate care precizează șirul dinaintea ultimei expresii regulate potrivite. Este un element nou în JavaScript 1.2.
- **rightContext:** Este o proprietate care precizează restul șirului, dincolo de ultima expresie regulată ce s-a potrivit. Este un element nou în JavaScript 1.2.
- **$1,..$9:** Este o proprietate care indică ultimele nouă subșiruri dintr-o potrivire; aceste subșiruri sunt incluse între paranteze. Este un element nou în JavaScript 1.2.

Obiectul expresie regulată

Obiectul expresie regulată conține șablonul unei expresii regulate.

Parametri

- **regexp:** Este un parametru care precizează numele obiectului expresie regulată. Este un element nou în JavaScript 1.2.
- **pattern:** Este un parametru care precizează textul expresiei regulate. Este un element nou în JavaScript 1.2.

Indicatori

- **i:** Este o opțiune care arată că în timpul căutării expresiei regulate, se ignoră dacă sunt utilizate majuscule sau minuscule.
- **g:** Este o opțiune care arată că în timpul căutării expresiei regulate, potrivirea (și căutarea) trebuie să fie globală.
- **gi:** Este o opțiune care arată că în timpul căutării expresiei regulate se ignoră dacă sunt utilizate majuscule sau minuscule și că potrivirea (și căutarea) trebuie să fie globală.

Proprietăți

- **global [true, false]:** Este o proprietate care stabilește valoarea indicatorului g în cod, adică dacă să fie o căutare globală (true) sau nu (false). Este un element nou în JavaScript 1.2.
- **ignoreCase [true, false]:** Este o proprietate care stabilește valoarea indicatorului i în cod, adică dacă să fie o căutare influențată de utilizarea majusculelor sau minusculelor (true) sau nu (false). Element nou în JavaScript 1.2.
- **lastIndex:** Este o proprietate (valoare întreagă) care indică poziția de la care va porni următoarea procedură de potrivire (de exemplu, lastIndex == 2). Este un element nou în JavaScript 1.2.

■ **source:** Este o proprietate (protejată la scriere) care conține textul șablonului. Element nou în JavaScript 1.2.

Metode

■ **compile:** Compilează expresia regulată. Această metodă este invocată de obicei la începutul scriptului, când expresia regulată este deja cunoscută și va rămâne constantă. Este un element nou în JavaScript 1.2.
■ **exec(str):** Execută o căutare a unei expresii regulate în șirul specificat (str). Este un element nou în JavaScript 1.2. Observație: folosește aceleași proprietăți ca și obiectul RegExp.
■ **test(str):** Execută o căutare a unei expresii regulate și a unui șir specificat (str). Este un element nou în JavaScript 1.2. Observație: folosește aceleași proprietăți ca și obiectul RegExp.

Obiectul reset [C | 2 | 3 | I]

Obiectul reset reprezintă un buton de reinițializare într-un formular HTML în JavaScript.

Proprietăți

■ **enabled:** Definește o valoare booleeană care precizează dacă butonul de reinițializare este activat. (Nu 2 | 3).
■ **form:** Definește o referire către obiectul form (formular) care conține butonul reset (reinițializare). (Nu 2 | 3).
■ **name:** Definește o valoare șir care conține numele elementului reset (reinițializare).
■ **value:** Definește o valoare șir care conține valoarea elementului reset (reinițializare).

Metode

■ **click():** Emulează acțiunea de executare a unui clic pe butonul de reinițializare.
■ **focus():** Precizează codului scris în JavaScript că trebuie să fie executat când butonul de reinițializare primește focusul. (Nu 2 | 3).

Elemente de tratare a evenimentelor

■ **onClick:** Precizează codului scris în JavaScript că trebuie să fie executat când se execută un clic pe butonul de reinițializare.
■ **onFocus:** Precizează codului scris în JavaScript că trebuie să fie executat când butonului de reinițializare i se atribuie focusul. (Nu 2 | 3).

Obiectul screen (Este un element nou în JavaScript 1.2)

Obiectul screen descrie (sau precizează) caracteristicile ecranului curent.

Proprietăți

■ **availHeight:** Este o proprietate care precizează înălțimea ecranului în pixeli (minus limitările impuse displayului static de către sistemul de operare). Este un element nou în JavaScript 1.2.
■ **availWidth:** Este o proprietate care precizează lățimea ecranului în pixeli (minus limitările impuse displayului static de către sistemul de operare). Este un element nou în JavaScript 1.2.
■ **height:** Este o proprietate care precizează înălțimea ecranului curent în pixeli. Este un element nou în JavaScript 1.2.

■ **width:** Este o proprietate care precizează lățimea ecranului în pixeli. Este un element nou în JavaScript 1.2.
■ **pixelDepth:** Este o proprietate care precizează numărul de biți (pe pixel) din ecranul curent. Este un element nou în JavaScript 1.2.
■ **colorDepth:** Este o proprietate care precizează numărul de culori posibile ce pot fi afișate în ecranul curent. Este un element nou în JavaScript 1.2.

Obiectul select [C | 2 | 3]

Obiectul select reprezintă o listă de selecție dintr-un formular HTML în JavaScript.

Proprietăți

■ **length:** Definește o valoare întreagă care conține numărul de opțiuni din lista de selecție.
■ **name:** Definește o valoare șir care conține numele listei de selecție.
■ **options:** Definește un tablou care conține fiecare dintre opțiunile din lista de selecție în ordinea în care apar. Proprietatea options are propriile ei proprietăți.

defaultSelected	Definește o valoare booleană care precizează dacă o opțiune a fost selectată prestabilit (dacă este, atunci ea reflectă atributul SELECTED).
index	Definește un număr întreg care reflectă indexul unei opțiuni.
length	Definește o coloană întreagă care reflectă numărul de opțiuni din lista de selecție.
name	Definește un șir care conține numele listei de selecție.
selected	Definește o valoare booleeană care precizează dacă opțiunea este selectată. Poate fi folosită pentru a selecta sau deselecta o opțiune.
selectedIndex	Definește un număr întreg care conține indexul opțiunii curente selectate.
text	Definește un șir care conține textul afișat în lista de selecție pentru o anumită opțiune.
value	Definește o valoare șir care precizează valoarea opțiunii specificate (adică, ea reflectă atributul VALUE).

■ **selectedIndex:** Reflectă indexul opțiunii curent selectate din lista de selecție.

Metode

■ **blur():** Această metodă retrage focusul din lista de selecție. (Nu 2 | 3).
■ **focus():** Această metodă atribuie focusul listei de selecție. (Nu 2 | 3).

Elemente de tratare a evenimentelor

■ **onBlur:** Precizează codului scris în JavaScript că trebuie să fie executat când lista de selecție pierde focusul.
■ **onFocus:** Precizează codului scris în JavaScript că trebuie să fie executat când lista de selecție primește focusul.
■ **onChange:** Precizează codului scris în JavaScript că trebuie să fie executat când se modifică opțiunea selectată din lista de selecție.

Obiectul string [C | 2 | 3 | I]

Obiectul string furnizează metode și proprietăți pentru lucrul cu literali și variabile de tip șir.

Proprietăți

- **length:** Definește o valoare întreagă care reprezintă lungimea șirului exprimată ca fiind numărul de caractere din șir.
- **prototype:** Furnizează un mecanism pentru adăugarea de proprietăți la un obiect string. (Nu 2).

Metode

- **anchor (*nume*):** Returnează un șir care conține valoarea obiectului string (șir) încadrat într-o etichetă container A, cu atributul NAME stabilit la valoarea *nume*.
- **big():** Returnează un șir care conține valoarea obiectului string încadrat într-o etichetă container BIG.
- **blink():** Returnează un șir care conține valoarea obiectului string încadrat într-o etichetă container BLINK.
- **bold():**Returnează un șir care conține valoarea obiectului string încadrat într-o etichetă container B.
- **charAt(*index*):** Returnează caracterul de la locația specificată de *index*.
- **charCodeAt(*index*):** Returnează un număr reprezentând o valoare din setul de coduri ISO –Latin-1 la poziția *index* curent. (Doar în Netscape Navigator 4.0 sau mai nou).
- **concat(*sir2*):** Combină două șiruri și obține un alt treilea șir din cele două. (Doar în Netscape Navigator 4.0 sau mai nou).
- **fixed():** Returnează un șir care conține valoarea obiectului string încadrat într-o etichetă container FIXED.
- **fontColor(*culoare*):** Returnează un șir care conține valoarea obiectului string încadrat într-o etichetă container FONT, cu atributul COLOR stabilit la valoarea *culoare*, care reprezintă numele unei culori sau al unui triplet RGB. (Nu I).
- **fontSize(*dimensiune*):** Returnează un șir care conține valoarea obiectului string încadrat într-o etichetă container FONTSIZE, cu mărimea stabilită la *dimensiune*. (Nu I).
- **fromCharCode(*num1, num2, ...*):** Returnează un șir format din caractere de tip ISO-Latin-1. Aceste caractere sunt precizate prin valorile lor de cod, care se scriu sub forma *num1, num2* etc.
- **indexOf(*sirGasit, startIndex*):** Returnează indexul primei apariții a lui *sirGasit*, căutarea pornind de la *startIndex*, care este o valoare opțională. Dacă nu este furnizată valoarea *startingIndex*, căutarea începe de la începutul șirului.
- **italics():** Returnează un șir care conține valoarea obiectului string (șir) încadrat într-o etichetă container I.
- **lasIndexOf(*sirGasit, startIndex*):** Returnează indexul ultimei apariții a lui *sirGasit*. Aceasta se realizează prin căutarea înapoi, începând de la *startIndex*. *startIndex* este opțional și se presupune că este ultimul caracter din șir, dacă nu se furnizează nici o valoare.
- **link(*href*):** Returnează un șir care conține valoarea obiectului string încadrat într-o etichetă container A, cu atributul HREF stabilit la valoarea *href*.
- **match(*expresie_regulata*):** Compară o expresie regulată cu un șir. Parametrul *expresie_regulata* reprezintă numele expresiei regulate, care poate fi exprimat fie ca o variabilă fie ca o literă.
- **replace(*expresie_regulata, newSubStr*):** Găsește și înlocuiește *expresie_regulata* cu valoarea newSubStr.

- **search(*expresie_regulata*):** Găsește expresia *expresie_regulata* și o compară cu un anumit șir.
- **slice(*inceput, [sfarsit]*):** Extrage o porțiune dintr-un șir dat și obține un nou șir din acea porțiune. *inceput* și *sfarsit* sunt amândouă indexuri care pornesc de la zero și pot fi folosite pentru a extrage primul, al doilea, al treilea caracter etc.
- **small():** Returnează un șir care conține valoarea obiectului string încadrat într-o etichetă container SMALL.
- **split(*separator*):** Returnează un tablou de șiruri create din divizarea șirului la fiecare apariție a lui *separator*. (Nu2 | I). Metoda split are o funcționalitate suplimentară în JavaScript cât și în browserul Netscape Navigator 4.0 sau mai nou. Această nouă funcționalitate include următoarele elemente:

Divizare fixă de șir și prin expresie regulată	Puteți divide acum șirul string atât conform argumentului expresie regulată, cât și printr-un șir fix.
Limitarea contorizării	Puteți adăuga acum o limită de contorizare pentru a preveni includerea elementelor goale în interiorul șirului.
Divizarea spațiilor libere	Reprezintă posibilitatea de divizare a unui spațiu liber (incluzând orice spațiu liber, cum ar fi blank, tab, linie nouă și altele).

- **strike():** Returnează un șir care conține valoarea obiectului string încadrat într-o etichetă container STRIKE.
- **sub():** Returnează un șir care conține valoarea obiectului string încadrat într-o etichetă container SUB.
- **substr(start, [lungime]):** Este folosită pentru a extrage un număr precizat (lungime) de caractere dintr-un șir. Folosiți start pentru a preciza locația de unde să înceapă acest proces de extragere. Este un element nou în Java Script 1.2.
- **substring (*primulIndex, ultimulIndex*):** Returnează un șir echivalent cu subșirul care începe de la *primulIndex* și se termină la caracterul dinainte de *ultimulIndex*. Dacă valoarea *primulIndex* este mai mare decât valoarea *ultimulIndex*, atunci șirul va începe la *ultimulIndex* și se va termina la caracterul dinainte de *primulIndex*. Observație: În JavaScript 1.2, x și y nu mai sunt interschimbate. Pentru a obține acest rezultat, trebuie să precizați în JavaScript 1.2 atributul language în interiorul etichetei <SCRIPT>.
- **sup():** Returnează un șir care conține valoarea obiectului string încadrată într-o etichetă container SUP.
- **toLowerCase():** Returnează un șir care conține valoarea obiectului string având toate caracterele transformate în minuscule.
- **toUpperCase():** Returnează un șir care conține valoarea obiectului string având toate caracterele transformate în majuscule.

Obiectul submit [C | 2 | 3 | I]

Obiectul submit reprezintă un buton de trimitere dintr-un formular HTML în JavaScript.

Proprietăți

- **enabled:** Definește o valoare booleeană care indică dacă butonul de trimitere este activat. (Nu 2 | 3).
- **form:** Este o referire la obiectul form care conține butonul de trimitere. (Nu 2 | 3).
- **name:** Definește o valoare șir care conține numele elementului butonului de trimitere.

- **type:** Definește o valoare șir care reprezintă atributul TYPE al etichetei <INPUT>. (Nu 2|1).
- **value:** Definește o valoare șir care conține valoarea elementului butonului de trimitere.

Metode

- **click():** Emulează acțiunea de executare a unui clic pe butonul de trimitere.
- **focus():** Atribuie focusul butonului de trimitere. (Nu 2|3).

Elemente de tratare a evenimentelor

- **onClick:** Precizează codului scris în JavaScript că trebuie să fie executat când se execută clic pe butonul de trimitere.
- **onFocus:** Precizează codului JavaScript că trebuie să fie executat când butonului de trimitere îi este atribuit focusul. (Nu2|3)

Obiectul text [C| 2|3|I]

Obiectul text reprezintă un câmp text dintr-un formular HTML în JavaScript.

Proprietăți

- **defaultValue:** Definește o valoare șir care conține valoarea prestabilită a elementului text (adică, valoarea atributului VALUE).
- **enabled:** Definește o valoare booleeană care arată când este activat câmpul text. (Nu 2|3).
- **form:** Este o referire la obiectul form care conține câmpul text. (Nu 2|3).
- **name:** Definește o valoare șir care conține numele elementului text.
- **type:** Definește o valoare șir care reprezintă atributul TYPE al etichetei <INPUT>. (Nu 2|1).
- **value:** Definește o valoare șir care conține valoarea elementului text.

Metode

- **gfocus():** Emulează acțiunea de acordare a focusului în câmpul text. (Nu 2|3).
- **blur():** Emulează acțiunea de retragere a focusului din câmpul text.
- **select():** Emulează acțiunea de selectare a textului în câmpul text.

Elemente de tratare a evenimentelor

- **onBlur:** Precizează codului scris în JavaScript că trebuie să fie executat când este retras focusul din câmp.
- **onChange:** Precizează codului scris în JavaScript că trebuie să fie executat când conținutul câmpului este modificat.
- **onFocus:** Precizează codului scris în JavaScript că trebuie să fie executat când câmpului îi este atribuit focusul.
- **onSelect:** Precizează codului scris în JavaScript că trebuie să fie executat când utilizatorul selectează o parte din text sau întreg textul din câmp.

Obiectul textarea [C|2|3|I]

Obiectul textarea reprezintă un câmp text multilinie dintr-un formular HTML în JavaScript.

Proprietăți

- **defaultValue:** Definește o valoare șir care conține valoarea prestabilită a elementului textarea (adică valoarea atributului VALUE).
- **enabled:** Definește o valoare booleeană care arată când este activat câmpul texarea. (Nu 2|3).

- **form:** Este o referire la obiectul form care conține câmpul de tip textarea. (Nu 2|3).
- **name:** Definește o valoare șir care conține numele elementului de tip textarea.
- **type:** Definește o valoare șir care reprezintă tipul obiectului textarea. (Nu 2|1).
- **value:** Definește o valoare șir care conține valoarea elementului textarea.

Metode

- **focus():** Emulează acțiunea de acordare a focusului în câmpul de tip textarea. (Nu 2|3).
- **blur():** Emulează acțiunea de retragere a focusului din câmpul de tip textarea.
- **select():** Emulează acțiunea de selectare a textului în câmpul de tip textarea.

Elemente de tratare a evenimentelor

- **onKeyUp:** Precizează codului scris în JavaScript că trebuie să fie executat când utilizatorul eliberează o anumită tastă. (Doar în Netscape Navigator 4.0). (4)
- **onKeyPress:** Precizează codului scris în JavaScript că trebuie să fie executat când utilizatorul ține apăsată o anumită tastă. (Doar în Netscape Navigator 4.0). (4)
- **onKeyDown:** Precizează codului scris în JavaScript că trebuie să fie executat când utilizatorul apasă o anumită tastă.(Doar în Netscape Navigator 4.0). (4)
- **onBlur:** Precizează codului scris în JavaScript că trebuie să fie executat când este retras focusul din câmp.
- **onChange:** Precizează codului scris în JavaScript că trebuie să fie executat când conținutul câmpului este modificat.
- **onFocus:** Precizează codului scris în JavaScript că trebuie să fie executat când câmpului îi este atribuit focusul.
- **onSelect:** Precizează codului scris în JavaScript că trebuie să fie executat când utilizatorul selectează o parte din text sau tot textul din câmp.

Obiectul window [C|2|3|I]

Obiectul window este obiectul de la nivelul cel mai de sus pentru fiecare fereastră sau cadru și obiectul părinte pentru obiectele document, location și history.

Proprietăți

- **defaultStatus:** Definește o valoare șir care conține valoarea prestabilită afișată în bara de stare.
- **frames:** Definește un tablou de obiecte pentru fiecare cadru dintr-o fereastră. Cadrele apar în tablou în ordinea în care apar și în codul sursă HTML.
- **innerHeight():** Precizează dimensiunea pe verticală a zonei de conținut (în pixeli). Este un element nou în JavaScript 1.2.
- **innerWidht():** Precizează dimensiunea pe orizontală a zonei de conținut (în pixeli). Este un element nou în JavaScript 1.2..
- **length:** Definește o valoare întreagă care indică numărul de cadre dintr-o fereastră părinte. (Nu I).
- **name:** Definește o valoare șir care conține numele ferestrei sau cadrului.
- **opener:** Este o referire la obiectul window care conține metoda open() folosită pentru a deschide fereastra curentă. (Nu 2|1).
- **pageXOffset:** Precizează poziția X curentă a zonei de fereastră vizibilă (exprimată în pixeli). Este un element nou în JavaScript 1.2.
- **pageYOffset:** Precizează poziția Y curentă a zonei de fereastră vizibilă (exprimată în pixeli). Este un element nou în JavaScript 1.2.
- **parent:** Definește un șir care indică numele ferestrei care conține setul de cadre.

- **personalbar [visible=true, false]:** Reprezintă bara de directoare din Netscape Navigator și precizează dacă este vizibilă sau nu. Este un element nou în JavaScript 1.2.
- **scrollbars [visible=true, false]:** Reprezintă barele de derulare a ferestrei curente și precizează dacă este vizibilă sau nu. Este un element nou în JavaScript 1.2.
- **self:** Este o alternativă pentru numele ferestrei curente.
- **status:** Este folosit pentru a afișa un mesaj în bara de stare, se realizează prin atribuirea unor valori acestei proprietăți.
- **statusbar=[true, false, 1,0]:** Precizează dacă bara de stare a ferestrei țintă este vizibilă sau nu.
- **toolbar=[true, false, 1,0]:** Precizează dacă bara de instrumente a ferestrei țintă este vizibilă sau nu.
- **top:** Este o alternativă pentru numele ferestrei de deasupra.
- **window:** Este o alternativă pentru numele ferestrei curente.

Metode

- **alert(*mesaj*):** Afișează *mesaj* într-o casetă de dialog.
- **back():** Trimite utilizatorul înapoi la adresa URL anterioară memorată în lista de istorice. (Simulează executarea unui clic pe butonul Back din Navigator.) Este un element nou în JavaScript 1.2.
- **blur():** Retrage focusul de pe fereastră. Pe multe sisteme, această metodă trimite fereastra în fundal. (Nu 2|I].
- **captureEvents():** Folosit într-o fereastră cu cadre (împreună cu metoda `enableExternalCapture`) și precizează că fereastra va captura toate evenimentele precizate.
- **clearInterval(*intervalID*):** Anulează intervalele de așteptare create cu metoda `setInterval`. Este un element nou în JavaScript 1.2.
- **close():** Închide fereastra. (Nu I).
- **confirm(*mesaj*):** Afișează *mesaj* într-o casetă de dialog cu butoane OK și cancel. Returnează `true` (adevărat) sau `false` (fals), în funcție de butonul apăsat de către utilizator.
- **disableExternalCapture():** Previne ca fereastra momentană cu cadre, să captureze evenimentele care apar în pagini încărcate dintr-o altă locație. Este un element nou în JavaScript 1.2.
- **enableExternalCapture():** Permite ferestrei curente (cu cadre) să captureze evenimente care apar în pagini încărcate dintr-o altă locație Este un element nou în JavaScript 1.2.
- **find([string], [true, false], [true, false]):** Găsește `string` în fereastra țintă. Există doi parametri true/false. Primul parametru precizează starea booleeană a diferențierii majusculelor de minuscule în timpul căutării, iar al doilea precizează dacă este făcută o căutare înapoi. Este un element nou în JavaScript 1.2.
- **focus():** Atribuie focusul ferestrei. Pe multe sisteme, aduce fereastra în prim plan. (Nu 2|I).
- **forward():** Trimite utilizatorul la următoarea adresă URL din lista de istorice. (Simulează executarea unui clic pe butonul Forward (înainte) din Navigator.) Este un element nou în JavaScript 1.2.
- **home():** Trimite utilizatorul la adresa paginii gazdă a utilizatorului. (Exemplu: într-o configurație prestabilită a browserului Netscape Navigator, trimite utilizatorul la `http://home.netscape.com`.) Este un element nou în JavaScript 1.2.
- **moveBy(orizontal, vertical):** Mută fereastra conform valorilor precizate pentru orizontală și verticală. Este un element nou în JavaScript 1.2.
- **moveTo(*x, y*):** Mută colțul din stânga sus al ferestrei la locația precizată; x și y sunt coordonatele ecranului. Este un element nou în JavaScript 1.2.

- **navigator(*url*):** Încarcă adresa *url* în fereastră. (Nu 2|3).
- **open (*url, nume, caracteristici*):** Deschide adresa *url* într-o fereastră numită *nume*. Dacă *name* nu există, atunci este creată o nouă fereastră cu acel nume. *Caracteristici* este un argument șir opțional, care conține o listă de caracteristici pentru noua fereastră. Lista de caracteristici conține oricare dintre următoarele perechi nume-valoare, separate prin virgule și fără spații suplimentare. (Nu I).

`toolbar=[yes,no,1,0]`	Precizează dacă fereastra ar trebui să aibă o bară de instrumente.
`location=[yes,no,1,0]`	Precizează dacă fereastra ar trebui să aibă un câmp de locație.
`directories=[yes,no,1,0]`	Precizează dacă fereastra ar trebui să aibă butoane de directoare.
`status=[yes,no,1,0]`	Precizează dacă fereastra ar trebui să aibă o bară de stare.
`menubar=[yes,no,1,0]`	Precizează dacă fereastra ar trebui să aibă meniuri.
`scrollbars=[yes,no,1,0]`	Precizează dacă fereastra ar trebui să aibă bare de derulare.
`resizable=[yes,no,1,0]`	Precizează dacă fereastra ar trebui să poată fi redimensionată.
`width= pixeli`	Precizează lățimea ferestrei în pixeli.
`alwaysLowered=[yes,no,1,2]`	Precizează (dacă e adevărat) că ferestra ar trebui să rămână sub alte ferestre. (Această proprietate are rezultate diferite pentru diferite sisteme de ferestre). Este un element nou în JavaScript 1.2. Observație: Scriptul trebuie să fie semnat pentru a folosi această caracteristică.
`alwaysRaised=[yes,no,1,2]`	Precizează (dacă e adevărat) că fereastra ar trebui să rămână deasupra. (Această proprietate are rezultate diferite pentru diferite sisteme de ferestre). Este un element nou în JavaScript 1.2. Observație: Scriptul trebuie să fie semnat pentru a folosi această caracteristică.
`dependent=[yes,no,1,2]`	Precizează că fereastra copil curentă se va închide când se închide fereastra părinte. Este un element nou în JavaScript 1.2.
`hotkeys=[yes,no,1,2]`	Precizează (dacă e adevărat) că majoritatea tastelor rapide sunt dezactivate în interiorul ferestrei curente. Este un element nou în JavaScript 1.2.
`innerWidth=pixeli`	Pecizează lățimea (în pixeli) a conținutului ferestrei curente. Este un element nou în JavaScript 1.2.
`innerHeight=pixeli`	Precizează înălțimea (în pixeli) a conținutului ferestrei curente. Este un element nou în JavaScript 1.2.
`outerWidth= pixeli`	Precizează lățimea bordurii exterioare orizontală a ferestrei curente. Este un element nou în JavaScript 1.2.

`outerHeight= pixeli`	Precizează înălțimea bordurii exterioare verticală a ferestrei curente. Este un element nou în JavaScript 1.2.
`screenX=pixeli`	Precizează distanța la care noua fereastră este plasată față de marginea din stânga a ecranului (pe orizontală). Este un element nou în JavaScript 1.2.
`screenY=pixeli`	Precizează distanța la care noua fereastră este plasată față de marginea de sus a ecranului (pe verticală). Este un element nou în JavaScript 1.2.
`z-lock=[yes, no, 1, 2]`	Precizează că fereastra curentă nu trece în ordinea ciclică z, așadar nu se plasează deasupra altor ferestre, chiar dacă este activată. Este un element nou în JavaScript 1.2. Observație: Scriptul trebuie să fie semnat pentru a folosi această caracteristică.
`height=pixeli`	Precizează înălțimea ferestrei în pixeli.

- **`print()`**: Tipărește conținutul unei ferestre sau al unui cadru. Este echivalentul acțiunii de apăsare a butonului Print din browserul Netscape Navigator. Este un element nou în JavaScript 1.2.
- **`prompt(mesaj, raspuns)`**: Afișează *`mesaj`* într-o casetă de dialog cu un câmp de introducere text, care are valoarea prestabilită *`raspuns`*. Răspunsul utilizatorului din câmpul de introducere de text este returnat sub forma unui șir.
- **`releaseEvents(eventType)`**: Precizează că fereastra curentă ar trebui să elibereze evenimentele și nu să le captureze, astfel că aceste evenimente pot fi trecute altor obiecte, probabil în continuarea ierarhiei evenimentelor. Este un element nou în JavaScript 1.2.
- **`resizeBy(orizontal,vertical)`**: Redimensionează fereastra, începând cu colțul din dreapta jos. Este un element nou în JavaScript 1.2.
- **`resizeTo(outerWidth, outerHeight)`**: Redimensionează fereastra, folosind proprietățile `outerHeight` și `outerWidth`. Este un element nou în JavaScript 1.2.
- **`routeEvent(event)`**: Trimite sau direcționează un eveniment trecând prin ierarhia normală a evenimentelor. Este un element nou în JavaScript 1.2.
- **`scrollBy(orizontal, vertical)`**: Derulează suprafața vizibilă a ferestrei curente cu valorile specificate. Este un element nou în JavaScript 1.2.
- **`scrollTo(x,y)`**: Derulează fereastra curentă la o poziție specificată, calculată la coordonatele x și y, începând din partea stângă-sus a ferestrei. Este un element nou în JavaScript 1.2.
- **`setInterval(functie, msec, [args])`**: Apelează în mod repetat o funcție după scurgerea perioadei specificate de parametrul `msec`. Este un element nou în JavaScript 1.2.
- **`setInterval(expresie, msec)`**: Evaluează *`expresie`* după scurgerea perioadei specificate de parametrul `msec`. Este un element nou în JavaScript 1.2.
- **`setTimeout(expresie, timp)`**: Evaluează *`expresie`* după scurgerea perioadei `timp`, care este o valoare dată în milisecunde. Timpul de așteptare poate fi denumit cu următoarea structură:

```
nume=setTimeout(expresie, timp)
```

- **`scrollTo(x,y)`**: Derulează fereastra curentă până la coordonatele x și y. (Nu 2|I).
- **`stop()`**: Oprește descărcarea curentă. Este echivalentul acțiunii de apăsare a butonului stop din browserul Netscape Navigator.
- **`clearTimeout(nume)`**: Anulează timpul de așteptare cu numele *`nume`*.

Elemente de tratare a evenimentelor

- **`onDragDrop`**: Precizează codului scris în JavaScript că trebuie să fie executat când utilizatorul pune un obiect în fereastră. (Doar în Netscape Navigator 4.0 sau mai nou.) (4.0).
- **`onBlur`**: Precizează codului scris în JavaScript că trebuie să fie executat când este retras focusul dintr-o fereastră. (Nu 2|I).
- **`onError`**: Precizează codului scris în JavaScript că trebuie să fie executat când apare o eroare în JavaScript în timp ce se încarcă un document. Poate fi folosit pentru a intercepta erorile JavaScript. Stabilirea acestui element de tratare a evenimentelor la valoarea `null` face ca erorile JavaScript să nu mai fie afișate utilizatorului. (Nu 2|I).
- **`onFocus`**: Precizează codului scris în JavaScript că trebuie să fie executat când ferestrei îi este atribuit focusul. (Nu 2|I).
- **`onLoad`**: Precizează codului scris în JavaScript că trebuie să fie executat când fereasta sau cadrul termină încărcarea.
- **`onMove`**: Precizează codului scris în JavaScript că trebuie să fie executat când utilizatorul mută o fereastră. (Doar în Netscape Navigator 4.0 sau mai nou.) (4.0).
- **`onResize`**: Precizează codului scris în JavaScript că trebuie să fie executat când utilizatorul redimensionează o fereastră.
- **`onUnload`**: Precizează codului scris în JavaScript că trebuie să fie executat când documentul din cadru sau dintr-o fereastră este închis.

Funcții independente, operatori, variabile și literali

Funcții independente

- **`escape(caracter)`**: Returnează un șir care conține codificarea ASCII a elementului *`caracter`* în forma %xx, unde xx reprezintă codificarea numerică a caracterului. (C|2|3|I).
- **`eval (expresie)`**: Returnează rezultatul evaluării elementului *`expresie`*, care este o expresie aritmetică. (C|2|3|I).
- **`isNaN (valoare)`**: Evaluează *`valoare`* pentru a vedea dacă este `NaN`. Returnează o valoare booleeană. (C|2|3|I). (Pe platformele UNIX, 2 nu este valabil).
- **`parseFloat(string)`**: Convertește *`string`* într-un număr în virgulă flotantă și returnează valoarea. Continuă să-l convertească până ce întâlnește un caracter nenumeric, apoi returnează rezultatul. Dacă primul caracter nu poate fi convertit într-un număr, funcția returnează *`NaN`* (zero pe platformele Windows). (C|2|3|I).
- **`parseInt(string, base)`**: Convertește *`string`* într-un număr întreg în baza *`base`* și returnează valoarea. Continuă să-l convertească până ce întâlnește un caracter nenumeric, apoi returnează rezultatul. Dacă primul caracter nu poate fi convertit într-un număr, funcția returnează *`NaN`* (zero pe platformele Windows). (C|2|3|I).
- **`taint(propertyName)`**: Adaugă posibilitatea modificării la *`propertyName`*. (C|3).
- **`toString()`**: Aceasta este o metodă pentru toate obiectele. Returnează obiectul sub formă de șir sau returnează `[object type]` dacă nu există o reprezentare sub formă de șir pentru obiect. (C|2|3). Observație: În JavaScript 1.2, convertește obiectele și șirurile în literali.
- **`unescape(string)`**: Returnează un caracter în funcție de codificarea ASCII din *`string`*. Codificarea ASCII ar trebui să fie de forma `%integer` sau `hexadecimalValue`. (C|2|3|I).
- **`untaint(propertyName)`**: Elimină posibilitatea modificării din *`propertyName`*. (C|3).

Instrucțiuni

- **`break`**: Termină o buclă `while` sau `for` și trece controlul programului la prima declarație care urmează după buclă. (2|3|4). Observație: În JavaScript 1.2, `break` are funcția suplimentară de a ieși din instrucțiunile cu etichetă.

■ **comment:** Este folosit pentru a adăuga un comentariu în interiorul scriptului. Acest comentariu este ignorat de browserul Netscape Navigator. Comentariile din JavaScript funcționează identic cu cele din limbajul C. Ele sunt incluse în structura /*(început),*/(sfârșit). (2|3|4).
■ **continue:** Termină execuția instrucțiunilor într-o buclă `while` sau `for` și continuă iterația buclei. (2|3|4). Observație: În JavaScript 1.2, `continue` are funcționalitatea suplimentară de a continua în cadrul instrucțiunilor cu etichetă.
■ **do while:** Configurează o buclă care continuă să execute instrucțiuni și cod până ce condiția este evaluată la `false`. Este un element nou în JavaScript 1.2.
■ **export:** Este folosită cu instrucțiunea `import`. În scripturile sigure, semnate, permite programatorului să exporte toate proprietățile, funcțiile și variabilele în alt script. Este un element nou în JavaScript 1.2.
■ **for([*expresie-initiala*]; [*conditie*]; [*expresie-incrementala*];):** Precizează deschiderea unei bucle `for`. Argumentele sunt: inițializează o variabilă (*expresie-initiala*), creează o condiție pentru testare (*conditie*) și precizează o expresie de incrementare (*expresie-incrementala*). (2|3|4).
■ **for…in:** Impune o variabilă tuturor proprietăților unui obiect și execută un bloc de cod pentru fiecare. (2|3|4).
■ **function [nume]():** Declară o funcție la care se poate referi sau poate fi accesată de elementele de tratare a evenimentelor (sau alte procese). (2|3|4).
■ **if…else:** Este o structură folosită pentru a testa dacă o anumită condiție este adevărată. Blocurile `if…else` pot conține instrucțiuni și funcții imbricate (și le pot apela), dacă o condiție este fie adevărată fie falsă. (2|3|4)
■ **import:** Este folosită cu instrucțiunea `export`. În scripturile sigure, semnate, permite programatorului să importe toate proprietățile, funcțiile și variabilele din alt script. Este un element nou în JavaScript 1.2.
■ **label (instructiuni cu eticheta):** Reprezintă instrucțiuni care creează o etichetă sau un indicator către cod, în altă parte în script. Prin apelarea acestei etichete, redirecționați scriptul către instrucțiunea etichetată.
■ **new:** Creează o instanțiere de obiect definit de utilizator. (Poate fi folosit de asemenea pentru a crea o instanțiere de obiecte încorporate, inerente în JavaScript, cum ar fi noul `Date`.) (2|3|4).
■ **return [valoare]:** Precizează o valoare care să fie returnată de o funcție dată. De exemplu, `return x` returnează valoarea variabilei asociată lui x. (2|3|4).
■ **switch:** Evaluează o expresie și încearcă să o potrivească unui șablon sau etichete `case`. Daca expresia se potrivește cu `case`, instrucțiunile ce urmează acelei etichete sunt executate. Este un element nou în JavaScript 1.2. (Operează similar cu instrucțiunea `switch` din sintaxa limbajului C).
■ **this:** Este o declarație folosită pentru a vă referi la un anumit obiect, așa cum e prezentat în acest exemplu: (2|3|4).

```
Onclick = 'javascript: my_function(this.form)'
```

■ **var [nume]:** Declară o variabilă după `nume`. (2|3|4).
■ **while:** Este o instrucțiune care începe o buclă `while`. Buclele `while` precizează că atât timp cât (while) o condiție este adevărată, se execută un anumit cod. (2|3|4).
■ **with:** Este o instrucțiune care stabilește valoarea pentru obiectul prestabilit, metodă similară cu crearea unei variabile globale cu ajutorul unei funcții. (2|3|4).

Operatori

■ **Operatori de atribuire:** vezi Tabelul E.1.

Tabelul E.1. Operatori de atribuire în JavaScript.

Operator	*Descriere*
=	Atribuie operandului din stânga valoarea operandului din dreapta.
+=	Adună operandul din stânga cu cel din dreapta și atribuie rezultatul operandului din stânga.
-=	Scade operandul din dreapta din operandul din stânga și atribuie rezultatul operandului din stânga.
*=	Înmulțește cei doi operanzi și atribuie rezultatul operandului din stânga.
/=	Împarte operandul din stânga la operandul din dreapta și atribuie rezultatul operandului din stânga.
%=	Împarte operandul din stânga la operandul din dreapta și atribuie restul operandului din stânga.

■ **Operatori aritmetici:** vezi Tabelul E.2.

Tabelul E.2. Operatori aritmetici în JavaScript.

Operator	*Descriere*
+	Adună operanzii din stânga și dreapta.
-	Scade operandul din dreapta din operandul din stânga.
*	Multiplică cei doi operanzi.
/	Împarte operandul din stânga la operandul din dreapta.
%	Împarte operandul din stânga la operandul din dreapta și evaluează restul.
++	Incrementează operandul cu 1 (poate fi folosit înainte sau după operand).
--	Decrementează operandul cu 1 (poate fi folosit înainte sau după operand).
-	Schimbă semnul operandului.

■ **Operatori pe bit:** Operatorii pe bit tratează operanzii lor ca numere binare, dar returnează valori numerice de tip JavaScript. (vezi Tabelul E.3).

Tabelul E.3. Operatori pe bit în JavaScript.

Operator	*Descriere*
AND (sau &)	Convertește operanzii în numere întregi pe 32 de biți, împerechează biții corespunzători și returnează 1 pentru fiecare pereche de 1. Returnează 0 pentru orice altă combinație.
OR (sau |)	Convertește operanzii în numere întregi pe 32 de biți, împerechează biții corespunzători și returnează 1 pentru fiecare pereche în care unul dintre biți este 1. Returnează 0 dacă ambii biți sunt zero.
XOR (sau ^)	Convertește operanzii în numere întregi pe 32 de biți, împerechează biții corespunzători și returnează 1 pentru fiecare pereche când doar unul dintre biți este 1. Returnează 0 la orice altă combinație.

(continuare)

Tabelul E.3. – continuare

Operator	*Descriere*
<<	Convertește operandul din stânga la un număr întreg pe 32 de biți și deplasează biții la stânga cu un număr de biți indicați de operandul din dreapta. Biții deplasați spre stânga care ies în afară sunt eliminați și sunt introduse zerouri dinspre dreapta.
>>>	Convertește operandul din stânga într-un număr întreg pe 32 de biți și deplasează biții la dreapta cu un număr de biți indicați de operandul din dreapta. Biții deplasați spre dreapta care ies în afară sunt eliminați și sunt introduse zerouri dinspre stânga.
>>	Convertește operandul din stânga la un număr întreg pe 32 de biți și deplasează biții la dreapta cu un număr de biți indicați de operandul din dreapta. Biții deplasați spre dreapta care ies în afară sunt eliminați, iar copiile bitului cel mai din dreapta sunt introduse dinspre stânga.

Operatori logici: vezi Tabelul E.4.

Tabelul E.4. Operatori logici în JavaScript.

Operator	*Descriere*
&&	„Și" logic. Returnează `true` (adevărat) când ambii operanzi au valoarea `true`; altfel, returnează `false`.
||	„Sau" logic. Returnează `true` (adevărat) când oricare dintre operanzi are valoarea `true`. Returnează `false` doar când ambii operanzi sunt `false`.
!	„Nu" logic. Returnează `true` (adevărat) dacă operandul este `false` (fals) și `false` (fals) dacă operandul este `true`. Acesta este un operator unar și precede operandul.

- **Operatori de comparație:** vezi Tabelul E.5.

Tabelul E.5. Operatori logici (de comparație) în JavaScript.

Operator	*Descriere*
==	Returnează `true` (adevărat) dacă operanzii sunt egali.
!=	Returnează `true` (adevărat) dacă operanzii nu sunt egali.
>	Returnează `true` (adevărat) dacă operandul din stânga este mai mare decât operandul din dreapta.
<	Returnează `true` (adevărat) dacă operandul din stânga este mai mic decât operandul din dreapta.
>=	Returnează `true` (adevărat) dacă operandul din stânga este mai mare sau egal cu operandul din dreapta.
<=	Returnează `true` (adevărat) dacă operandul din stânga este mai mic sau egal cu operandul din dreapta.

- **Operatori condiționali:** Expresiile condiționale au o singură formă:

(conditie) ? val1 : val2

Când *conditie* are valoarea `true` (adevărat), expresia este evaluată la *val1*, altfel este evaluată la *val2*.

- **Operatori pentru șiruri:** Operatorul de concatenare (+) este unul dintre cei doi operatori pentru șiruri. El evaluează un șir, combinând operanzii din stânga și dreapta. Operatorul de atribuire și concatenare (+=) este de asemenea disponibil.
- **Operatorul `typeof`:** Acest operator returnează tipul unicului său operand. Tipurile posibile sunt `object` (obiect), `string` (șir), `boolean`, `function` (funcție) și `undefined` (nedefinit).
- **Operatorul `void`:** Operatorul `void` consideră o expresie ca un operand, dar nu returnează nici o valoare.
- **Precedența operatorilor:** JavaScript aplică regulile de precedență a operatorilor, după cum urmează (de la precedenta cea mai mică până la cea mai mare).

Virgulă (,)
Operatori de atribuire (=, +=, - =, *=, /=, %=)
Condițional (? :)
SAU logic (||)
ȘI logic (&&)
SAU pe biți (|)
XOR pe biți (^)
ȘI pe biți (&)
Egalitate (= =, !=)
Relațional (<, <=, >, >=)
Deplasare (<<, >>, >>>)
Adunare/scădere (+, -)
Multiplicare/împărțire/modulo (*, /, %)
Negare/increment (!, -, ++, - -)
Apelare, membru (( ), [ ])

Anexa F
VBScript

Această anexă conține un rezumat al operatorilor, instrucțiunilor și funcțiilor folosite în Visual Basic Scripting Edition.

Categorie /cuvânt cheie	*Tip*	*Utilizare*
		Matematică
`Atn`	Funcție	Returnează arctangenta unui număr `Atn (numar)`
`Cos`	Funcție	Returnează cosinusul unui unghi `Cos (numar)`
`Exp`	Funcție	Returnează un număr ridicat la o putere `Exp (numar)`
`Log`	Funcție	Returnează logaritmul unui număr `Log (numar)`
`Randomize`	Instrucțiune	Activează generatorul intern de numere aleatoare `Randomize`
`Rnd`	Funcție	Returnează un număr aleator `Rnd`
`Sin`	Funcție	Returnează sinusul unui unghi `Sin(numar)`
`Sqr`	Funcție	Returnează rădăcina pătrată a unui număr `Sqr(numar)`
`Tan`	Funcție	Returnează tangenta unui unghi `Tan(numar)`
		Tratarea tablourilor
`Dim`	Instrucțiune	Declară un tablou `Dim numetablou([indici])`
`Erase`	Instrucțiune	Șterge conținutul unui tablou `Erase numetablou`
`IsArray`	Funcție	Returnează `True` (adevărat) dacă *`var`* este un tablou și `False` (fals) în caz contrar `IsArray(var)`
`LBound`	Funcție	În VBScript, returnează întotdeauna valoarea 0 `Lbound(numetablou)`
`Preserve`	Instrucțiune	Copiază conținutul unui tablou dinamic într-un tablou dinamic redimensionat `Redim Preserve numetablou(indici)`

Categorie /cuvânt cheie	*Tip*	*Utilizare*

`ReDim`	Instrucțiune	Declară un tablou dinamic sau redimensionează un tablou dinamic (vedeți `Preserve`)
		`ReDim numetablou ()`
		sau
		`ReDim numetablou ([indici])`

`UBound`	Instrucțiune	Returnează cel mai mare indice al unui tablou `Ubound (numetablou)`
		Atribuire
`=`	Operator	Atribuie o valoare unei variabile sau proprietăți `variabila = valoare`
`Set`	Instrucțiune	Atribuie o referință la un obiect unei variabile `Set variable = object`
		Comentariu
`Rem`	Instrucțiune	Declară linia care urmează ca un comentariu ce trebuie ignorat de către motorul limbajului `Rem comment_text`
		Constante / Literali
`Empty`	Literal	Declară o valoare de variabilă neinițializată specială `variabila = Empty`
`False`	Constantă	Declară o valoare booleeană reprezentând valoarea 0 `variabila = False`
`Nothing`	Literal	Folosit pentru a dezasocia o referință la un obiecst de o variabilă; folosit cu `Set` `Set variabila = Nothing`
`Null`	Literal	Nu reprezintă nici o dată validată `variabila = Null`
`True`	Constantă	Valoare booleeană reprezentând valoarea - 1 `variabila = True`
		Conversii
`Abs`	Funcție	Returnează valoarea fără semn (absolută) a numărului `Abs(numar)`
`Asc`	Funcție	Returnează codul ANSI/ASCII al caracterul `Asc(sir)`

(continua

Categorie /cuvânt cheie	Tip	Utilizare
`CBool`	Funcție	Returnează o valoare `Variant` de subtip booleeană din orice expresie validă `CBool(expresie)`
`CByte`	Funcție	Returnează o valoare `Variant` de subtip Byte (bit) din orice expresie validă `CByte(expresie)`
`CDate`	Funcție	Returnează o valoare `Variant` de subtip Date (dată) din orice expresie de date validă `CDate(expresie)`
`CDbl`	Funcție	Returnează o valoare `Variant` de subtip Double Precision (dublă precizie) din orice expresie numerică validă `CDbl(expresie)`
`Chr`	Funcție	Returnează caracterul corespunzător codului ANSI sau ASCII `Chr(numar)`
`CInt`	Funcție	Returnează o valoare `Variant` de subtip Integer (întreagă) din orice expresie numerică validă. `CInt(expresie)`
`CLng`	Funcție	Returnează o valoare `Variant` de subtip Long Integer (întreagă lungă) din orice expresie numerică validă `CLng(expresie)`
`CSng`	Funcție	Returnează o valoare `Variant` de subtip Single Precision (precizie simplă) din orice expresie numerică validă `CSng(expresie)`
`CStr`	Funcție	Returnează o valoare `Variant` de subtip String (șir) din orice expresie validă `CStr(expresie)`
`DateSerial`	Funcție	Returnează o valoare `Variant` de subtip dată din valorile `an`, `luna` și `zi` valide `DateSerial(an, luna, zi)`
`DateValue`	Funcție	Returnează o valoare `Variant` de subtip Date (dată) din orice expresie validă `DateValue(expresion)`
`Hex`	Funcție	Returnează o valoare `Variant` de subtip șir reprezentând valoarea hexazecimală a unui număr `Hex(numar)`

Categorie /cuvânt cheie	Tip	Utilizare
`Int`	Funcție	Returnează o valoare `Variant` de subtip întreg rotunjit în minus față de numărul dat `Int(numar)`
`Fix`	Funcție	Returnează o valoare `Variant` de subtip întreg rotunjit în plus față de numărul dat `Fix(numar)`
`Oct`	Funcție	Returnează o valoare `Variant` de subtip șir reprezentând valoarea octală a unui număr `Oct(numar)`
`Sgn`	Funcție	Returnează o valoare `Variant` de subtip întreg reprezentând semnul unui număr `Sgn(numar)` Valorile > 0 returnează 1 Valorile = 0 returnează 0 Valorile < 0 returnează –1
`TimeSerial`	Funcție	Returnează o valoare `Variant` de subtip Date (dată) din valorile `ora`, `minute` și `secunde` valide `TimeSerial(ora, minute, secunde)`
`TimeValue`	Funcție	Returnează o valoare `Variant` subtip Date (dată), din orice expresie de timp validă. `TimeValue(expresie)`
		Data și ora
`Date`	Funcție	Returnează data curentă a sistemului `Date()`
`DateSerial`	Funcție	Returnează o valoare `Variant` de subtip Date (dată) din valorile an, lună și zi, valide `DateSerial(an, luna, zi)`
`DateValue`	Funcție	Returnează o valoare `Variant` de subtip Date (dată) din orice expresie de date validă `DateValue (expresie)`
`Day`	Funcție	Returnează o valoare `Variant` de subtip Integer (întreg) reprezentând ziua (`1-31`), dintr-o expresie de dată validă `Day (expresiedata)`
`Hour`	Funcție	Returnează o valoare `Variant` de subtip Integer (întreg) reprezentând ora (`0-23`), dintr-o expresie de oră validă `Hour (expresieora)`

(continuare)

Categorie /cuvânt cheie	Tip	Utilizare
Minute	Funcție	Returnează o valoare Variant de subtip Integer (întreg) reprezentând minutele (0-60) dintr-o expresie de oră validă Minute (*expresieora*)
Month	Funcție	Returnează o valoare Variant de subtip Integer (întreg) reprezentând luna (1-12) dintr-o expresie de dată validă Month (*expresiedata*)
Now	Funcție	Returnează data și ora sistemului Now ()
Second	Funcție	Returnează o valoare Variant de subtip Integer (întreg) reprezentând secundele (0-60) dintr-o expresie de oră validă Second(*expresieora*)
Time	Funcție	Returnează ora curentă a sistemului Time ()
TimeSerial	Funcție	Returnează o valoare Variant de subtip Date (dată) din valorile oră, minute și secunde valide TimeSerial(*ora, minute, secunde*)
TimeValue	Funcție	Returnează o valoare Variant de subtip Date (dată), din orice expresie de timp validă TimeValue(*expresie*)
Weekday	Funcție	Returnează o valoare Variant de subtip Integer (întreg) între 1 și 7, reprezentând ziua din săptămână, începând cu duminică (Sunday), dintr-o expresie de dată Weekday(*expresiedata*)
Year	Funcție	Returnează o valoare Variant de subtip Integer (întreg) reprezentând anul dintr-o expresie de dată validă Year(*expresiedata*)
		Declarații
Dim	Instrucțiune	Declară o variabilă Dim *variabila*
End	Instrucțiune	Declară sfârșitul unei proceduri sau funcții Sub End Sub End Function

Categorie /cuvânt cheie	Tip	Utilizare
Exit	Instrucțiune	Folosit cu instrucțiunile Do, For, Function sau Sub pentru a părăsi rutina înainte de sfârșit Exit Do/For/Function/Sub
Function	Instrucțiune	Declară o funcție și lista de argumente transmisă în funcție și declară sfârșitul unei funcții. Este folosit de asemenea cu instrucțiunea Exit pentru a termina o funcție înainte de sfârșit Function *numefunctie(listaargumente)* Exit Function End Function Public *variabila*
Sub	Instrucțiune	Declară o procedură de utilizator sau element de tratare a evenimentelor și lista de argumente, dacă există, și declară sfârșitul unei proceduri de utilizator și al tratării unui eveniment. Este folosită de asemenea cu Exit pentru a termina o procedură de utilizator sau un element de tratare a evenimentelor, înainte de sfârșit Sub *numesubrutina([listaargumente])* Exit Sub End Sub
		Tratarea erorilor
Clear	Metodă	Este o metodă a obiectului Err de a reinițializa la valoarea 0 proprietatea Err.Number Err.Clear
Description	Proprietate	Este o proprietate a obiectului Err, care conține o descriere a ultimei erori, după cum este precizat în proprietatea Err.Number Err.Description
Err	Obiect	Definește un obiect conținând informația referitoare la ultima eroare Err.*proprietate* | *metoda*
On Error	Instrucțiune	Este folosită cu Resume Next pentru a continua executarea cu linia următoare imediat după linia în care a apărut eroarea On Error Resume Next
Raise	Metodă	Este o metodă a obiectului Err folosită pentru a simula apariția unei erori specificată de număr Err.Raise (*numareroare*)

(continuare)

Categorie /cuvânt cheie	Tip	Utilizare
`Number`	Proprietate	Este o proprietate a obiectului `Err`, care conține codul de eroare pentru ultima eroare sau valoarea 0 dacă nu au apărut erori `Err.Number`
`Source`	Proprietate	Returnează numele obiectului sau aplicației care au declanșat eroarea `Err.Source`
		Intrare / Ieșire
`InputBox`	Funcție	Afișează o casetă de dialog pentru a permite o intrare a utilizatorului `InputBox (legenda[,titlu][,valoare][,x][,y])`
`MsgBox`	Funcție	Afișează o casetă de dialog `MsgBox (prompt[, definition][, title])`
		Operatori
`+`	Operator	Sumă de două expresii numerice `result = expr1 + expr2`
`And`	Operator	Operator de conjuncție logică `If expresie AND expresie Then`
`/`	Operator	Operator de împărțire `result = expresie / expresie`
`=`	Operator	Operator de egalitate `If expresie = expresie Then`
`Eqv`	Operator	Operator de echivalență logică `If expresie Eqv expresie Then`
`^`	Operator	Operator exponențial `result = expresie ^ expresie`
`>`	Operator	Comparația mai mare decât `If expresie > expresie Then`
`>=`	Operator	Comparația mai mare sau egal cu `If expresie >= expresie Then`
`Imp`	Operator	Implicație logică `If expresie Imp expresie Then`
`<>`	Operator	Comparația de inegalitate `If expresie < > expresie Then`
`\`	Operator	Operator de împărțire întreagă `result = expresie \ expresie`

Categorie /cuvânt cheie	Tip	Utilizare
`<`	Operator	Comparația mai mic decât `If expresie < expresie Then`
`<=`	Operator	Comparația mai mic sau egal cu `If expresie <= expresie Then`
`Mod`	Operator	Modulo, returnează doar restul unei împărțiri între două numere. `result = expresie mod expresie`
`*`	Operator	Înmulțire `result = expresie * expresie`
`-`	Operator	Scădere `result = expresie - expresie`
`Or`	Operator	Disjuncție logică `If expresie Or expresie Then`
`&`	Operator	Concatenare a două valori șir `result = sir & sir`
`Xor`	Operator	Excluziune logică `If expresie Xor expresie Then`
		Opțiuni
`Option Explicit`	Instrucțiune	Forțează o eroare la momentul compilării dacă este găsită o variabilă nedeclarată `Option Explicit`
		Derularea programului

`Call`	Instrucțiune	Transferă execuția către o subrutină sau un element de tratare a evenimentelor. De asemenea, poate fi folosit pentru a răspunde la acțiunile utilizatorului.
		`Call myroutine()`
		`Call cmdbutton_OnClick()`

`Do ... Loop`	Instrucțiune	Repetă codul până ce este îndeplinită o condiție.
		`Do While conditie`
		`...`
		`Loop`
		sau
		`Do Until conditie`
		`...`
		`Loop`
		sau
		`Do`
		`...`
		`Loop While conditie`

(continuare)

Categorie /cuvânt cheie	Tip	Utilizare

		sau
		`Do`
		`...`
		`Loop Until conditie`

`For ... Next`	Instrucțiune	Repetă un bloc de cod până ce contorul ajunge la un număr dat.
		`For contor = mic to mare [pas]`
		`...`
		`Next`

`If ... Then ... Else`	Instrucțiune	Execuție condițională a codului
		`If conditie Then`
		`... (if conditie met)`
		`Else`
		`... (if conditie not met)`
		`End If`

`Select Case`	Instrucțiune	Execuție selectivă a codului; *expresietest* trebuie să se potrivească cu *expresie*
		`Select Case expresietest`
		`Case expresie`
		`...`
		`Case expresie`
		`...`
		`Case Else`
		`End Select`

`While ...Wend`	Instrucțiune	Execuă un bloc de cod până ce este îndeplinită o condiție.
		`While expresie`
		`...`
		`Wend`

		Șiruri
`InStr`	Funcție	Returnează punctul de pornire al unui șir în cadrul altui șir sau `0` dacă nu este găsit `result = InStr (start, cautat, gasit)`
`LCase`	Funcție	Convertește un șir în minuscule `result = Lcase(sir)`
`Left`	Funcție	Returnează *n* caractere cele mai din stânga dintr-un șir `result = Left(sir, length)`
`Len`	Funcție	Returnează lungimea unui șir `result = Len(sir)`

Categorie /cuvânt cheie	Tip	Utilizare
`LTrim`	Funcție	Elimină toate spațiile libere de la începutul șirului `result = LTrim(sir)`
`Mid`	Funcție	Returnează un șir de lungime `L`, începând de la `S` din *sir* `result = Mid(sir, S, L)`
`Right`	Funcție	Returnează *n* caractere cele mai din dreapta dintr-un șir `result = Right(sir, n)`
`RTrim`	Funcție	Elimină toate spațiile libere de la sfârșitul unui șir `result = RTrim(sir)`
`Space`	Funcție	Returnează un șir cu *n* spații `result = Space(n)`
`StrComp`	Funcție	Returnează o valoare `Variant` de subtip întreg, reprezentând rezultatul unei comparații între două șiruri. `result = StrComp(sir1, sir2)` *sir1* < *sir2* returneaza *–1* *sir1* < *sir2* returneaza *0* *sir1* < *sir2* returneaza *1*
`String`	Funcție	Returnează un șir format din caracterul `C`, de lungime `L` `result = String(L, C)`
`Trim`	Funcție	Elimină atât spațiile de la începutul șirului, cât și spațiile de la sfârșit `result = Trim(sir)`
`UCase`	Funcție	Returnează un șir format din caractere scrise cu majuscule `result = UCase(sir)`
		Variante
`IsArray`	Funcție	Returnează `True` (adevărat) (-1) dacă *expresie* este un tablou și `False` (fals) (0) în caz contrar `result = IsArray(expresie)`
`IsDate`	Funcție	Returnează `True` (-1) dacă *expresie* este o dată validă și `False` (0) în caz contrar `result = IsDate(expresie)`
`IsEmpty`	Funcție	Returnează `True` (-1) dacă *expresie* este egală cu un subtip `Empty` și `False` în caz contrar `result = IsEmpty(expresie)`

Categorie /cuvânt cheie	*Tip*	*Utilizare*
`IsNull`	Funcție	Returnează `True` (-1) dacă *`expresie`* este egală cu un subtip `Null` și `False` în caz contrar *`result`* = `IsNull(`*`expresie`*`)`
`IsNumeric`	Funcție	Returnează `True` (-1) dacă `expresie` este o expresie numerică validă și 0 în caz contrar *`result`* = `IsNumeric(`*`expresie`*`)`
`VarType`	Funcție	Returnează un număr întreg reprezentând subtipul datei `Variant` *`result`* = `VarType(`*`expresie`*`)`

Anexa G
Resurse pentru dezvoltare

de Lee Anne Phillips

În această anexă

- **Browsere Web 882**
- **Editoare HTML 888**
- **Instrumente pentru situri Web: Publicarea pe Web și administratori de sit, hărți de situri, diverse produse shareware 892**
- **Grafică 899**
- **Programe plug-in din diverse surse 905**
- **Multimedia pentru profesioniști și utilizări ocazionale 906**
- **Limbaje de scriptare/independente de platformă 910**
- **Validatori HTML 915**
- **Servere Web 917**
- **Common Gateway Interface (CGI) 921**
- **Firewall, servere proxy și securitate 926**
- **Biblioteci și alte resurse 929**
- **Observații pentru studii viitoare 940**

Dacă vă așteptați să găsiți în această anexă o listă care să conțină totul, fără a fi separat „grâul de neghină", ei bine, nu veți găsi. Această anexă este de fapt o listă tematică, cu informații detaliate despre cele mai bune programe software disponibile pentru crearea paginilor Web, precum și cu liste conținând serverele cele mai folosite și alte „infrastructuri", realizând o referință cu însemnări și o bibliografie Web, acoperind dacă nu toată, măcar o mare parte a acestei cărți. Ca orice lucru important din viață, scrisul reprezintă un efort cooperant, implicând mai multe persoane, unele dintre aceste persoane fiind trecute pe pagina de început a acestei cărți. Doresc să le mulțumesc în special lui David și Rhonda Crowder, lui Robert McDaniel și lui Kelly Murdock, care mi-au spus unele dintre adresele lor URL favorite pentru a face această anexă mai utilă, cât și Lisei Lord care m-a ajutat să o fac mai lizibilă.

Am împărțit lista în mod arbitrar, în următoarele secțiuni, deși unele dintre ele se suprapun puțin.

- Browsere Web
- Editoare HTML
- Instrumente pentru situri Web: publicarea pe Web, administratori de sit, hărți de situri, diverse produse shareware
- Grafică
- Programe plug-in din diverse surse
- Multimedia pentru profesioniști și utilizări ocazionale

- Limbaje de scriptare/independente de platformă
- Validatori HTML
- Servere Web
- Common Gateway Interface (CGI)
- Firewall, servere proxy și securitate
- Biblioteci și alte resurse
- Observații pentru studii viitoare, incluzând motoarele de căutare

Unele dintre următoarele capitole sunt împărțite în subcategorii pentru a putea facilita organizarea mai multor date într-un singur loc.

Browsere Web

Browserele Web sunt inima rețelei Web și, la nivel de mașină, justificarea existenței serverelor HTTP și a celorlalte tehnologii complicate ce afișează conținutul paginilor pe ecranul dumneavoastră. Ele transformă biții și octeții bruți electronici în cuvinte, imagini, sunet sau alte lucruri pe care le puteți înțelege. Ele sunt, de asemenea, servitorii și intermediarii dumneavoastră, alegând automat cea mai bună modalitate de a obține și a reda o anumită informație, ascunzând în același timp detaliile supărătoare ce v-ar putea deranja.

Este o muncă importantă, iar atenția arătată de producătorii principali de software arată importanța care crede că o vor avea browserele Web în viitor. Tabelul G.1 oferă o selecție a browserelor, nu doar a celor mai populare, dar și a celor care se adresează utilizatorilor cu nevoi specifice, nesatisfăcuți întotdeauna de browserele renumite.

Tabelul G.1. Browsere pentru HTML.

Produs	*Tip/Suport*	*Cerințele minime ale sistemului*	*Nivelul de preț*	*Platforme*
Microsoft Internet Explorer 3.0	Grafic HTML 3.2 plus extensiile	Medii	Gratis, până la $$	Win95 Mac WinNT
Microsoft Internet Explorer 4.0	Grafic HTML 4.0 plus extensiile	Medii	Gratis, până la $$	Win95
Netscape Navigator 3.0	Grafic HTML 3.2 plus extensiile	Medii	Gratis, până la $$	Win95 Windows Mac/UNIX
Netscape Navigator 4.0	Grafic HTML 3.2 plus extensiile, plus o parte din HTML 4.0	Medii	Gratis, până la $$	Win95 Mac UNIX
NCSA Mosaic 3.0	Grafic HTML 3.0	Medii	Gratis	Win95 Windows
Opera3.0	Grafic HTML 3.2	Scăzute	$$	Win95 Windows

Produs	*Tip/Suport*	*Cerințele minime ale sistemului*	*Nivelul de preț*	*Platforme*
Tango	Grafic Multilimbaj HTML 3.2	Scăzute	$$	Win95/NT Windows
Lynx	Mod text HTML 3.2*	Scăzute	Gratis	UNIX DOS
Emacs W3	Grafic/ Mod text HTML 3.2*	Scăzute	Gratis	UNIX Win95 Mac
Ariadna	Grafic Rus HTML 3.2	Medii	$$	Win95/NT
Grail	Grafic Extensibil HTML 3.2	Medii	Gratis	UNIX Mac Win95/NT
HotJava	Grafic HTML 3.2	Medii	Gratis	Oricare
Amaya	Grafic HTML 3.2+	Medii	Gratis	UNIX Win95/NT
Cyberdog	Grafic	Medii	$$$	Mac
Cello	Grafic	Scăzute	Gratis	Windows

OBSERVAȚIE

În toate tabelele din această anexă, calculatoarele sunt descrise în termenii echivalenței lor cu Intel, deoarece majoritatea utilizatorilor par să știe cum se situează calculatorul lor relativ la standardul omniprezent:

Ridicat: Stație de lucru Pentium Pro, server sau echivalent

Mediu: 486 performant, Pentium sau echivalent

Scăzut: 386 până la 486 cu performanțe mari sau echivalent

Semnele dolar ($) sunt grosier echivalate cu cifrele din notarea zecimală, dar puțin exagerate, gândindu-ne la efectul pe care l-ar avea asupra buzunarului utilizatorului.

$$: Mai puțin de 75$

$$$: În jur de 100 $ sau mai mult

$$$$: În jur de 1000 $

$$$$$: 5000 $ sau mai mult

Microsoft Internet Explorer 4.0

`http://www.microsoft.com/ie/`

O dată cu lansarea browserului Internet Explorer 4.0, Microsoft a făcut un pas uriaș spre integrarea lui Internet Explorer în sistemul de operare. De fapt, datorită integrării puternice în suprafața de lucru Windows 95+, „upgrade" ar putea fi descris ca fiind în fapt

înlocuire a sistemului de operare. Implementările sinuoase ale etichetelor standard, cât și tehnologiile de susținere a lor au făcut ca firma Microsoft să aibă o reputație de „băiat rău", dar reprezintă o forță ce trebuie recunoscută, datorită controlului pe care îl are asupra unei mări părți din piața calculatoarelor. Dacă ați încercat vreodată să scăpați de Windows 95 Inbox, dorind să folosiți în locul lui Microsoft Exchange (și MSN), atunci cunoașteți ce înseamnă cu adevărat cuvântul „persistență". Ultima versiune are multe elemente de compatibilitate, deși implementarea limbajului Java și JavaScript, încă lasă de dorit. De fapt, implementarea limbajului Dynamic HTML este mai apropiată de proiectul de standard de la W3C, decât orice alt browser. Acest browser reprezintă o alegere bună pentru birourile bazate pe platforma Microsoft și pe rețele intranet, dar Microsoft are o compatibilitate mai mică cu Macintosh și aproape nici una cu UNIX; așadar siturile care acceptă mai multe platforme ar trebui să rețină acest aspect.

Internet Explorer a avansat la nivelul de adevărat inovator, iar integrarea sa în suprafața de lucru a mediului Windows este cam la nivelul următor după nimic. Nici o altă platformă nu vă oferă atât de mult acces la Visual Basic, VBScript, ActiveX și alte instrumente Microsoft, ceea ce poate fi bineînțeles și o sabie cu două tăișuri. Pentru rețelele intranet care funcționează pe platforme Microsoft și Intel, există o integrare completă cu multe aplicații mai vechi, folosind metode de acces particulare. Pentu programator, există acces instantaneu la biblioteci extinse de cod și controale deja scrise pentru a putea rula aproape orice funcție, cod care ar fi trebuit să fie scris de programator de la început și făcut să meargă pe orice altă platformă.

Pe de altă parte, când clienții dumneavoastră sunt pe Internet, nu puteți depinde de platformele Windows disponibile, așa că multe dintre instrumente și controale au o importanță mai mică sau trebuie să fie făcute disponibile în variante alternative, aceasta reprezentând o posibilă dublare a efortului. Orice dezvoltare a unui proiect mare pentru Internet, trebuie să țină cont atât de Internet Explorer, cât și de Netscape Navigator, dar, pe termen scurt, eforturile pentru „80% din funcționalitate" ar putea fi puternic susținute de folosirea browserului Internet Explorer ca instrument pentru prototipuri, astfel fiecare proiect putând beneficia de folosirea judicioasă a instrumentelor WinNT și Internet Explorer pe partea de server, dacă există o bază de date sau alte resurse pe acea platformă.

Pe piața actuală, Internet Explorer împarte cu Netscape Navigator onoarea de a fi browserele care oferă în premieră cele mai noi și mai importante facilități, inclusiv Dynamic HTML (cod care se poate rescrie el însuși, pe loc, fără intervenția serverului), realitate virtuală, scriptare și alte funcții pe partea clientului, multimedia și toate celelalte surprize care fac să bată mai repede inima programatorilor pentru Web. Această carte se concentrează mai ales pe ce puteți face atât cu Netscape Navigator, cât și cu Internet Explorer, așa că o scurtă cercetare a cuprinsului vă va oferi o mai bună privire de ansamblu asupra capacităților lor, decât vă pot oferi eu în acest spațiu limitat.

Rulează în special pe platformele Win95NT, dar acceptă și platformele Mac, de obicei cu seturi de caracteristici diferite.

Netscape Navigator 4.0

`http://www.netscape.com/`

Marele bunic al creatorilor de pagini Web comerciale și acoperind încă de departe cel mai mare segment din piață și suport de situri Web, Netscape a fost un inovator consecvent tot timpul. Deși criticat pe drept pentru multe din extensiile pe care le-a făcut disponibile pentru creatorii de pagini HTML, pentru o perioadă de timp a condus procesul dezvoltării de standarde prin satisfacerea cerințelor utilizatorilor, iar extensiile sale au fost proiectate

pentru a crea foarte puține probleme pentru alte browsere. Incursiunea sa inițială în Dynamic HTML, implementat cu o etichetă particulară <LAYER>, cât și foile de stil aprobate, reprezintă, ca de obicei pentru Netscape Navigator, o vâslire împotriva curentului, dar este foarte ușor de lucrat cu el, comparativ cu modelul de obiect de la W3C. Această versiune va fi completată cu o implementare a standardului W3C pe măsură ce limbajul HTML 4.0 se va dezvolta.

Netscape Navigator s-a modificat de-a lungul timpului până aproape de a fi de nerecunoscut, de la o simplă colecție de câteva zeci de etichete și afișări în nuanțe de gri plicticoase care existau acum câțiva ani în urmă. Creatorul de pagini Web al zilelor noastre trebuie să stăpânească o mulțime amețitoare de tehnologii noi, scriptare interactivă, programare pe partea clientului, secvențe ce pot fi descărcate, scrise în limbajul Java sau alte limbaje, multimedia, realitate virtuală, marcaj extensibil (în curând) și alte metode pentru a face ca paginile să fie la înălțime. Aceasta este atât o facilitate, cât și o problemă. În multe cazuri, crearea paginilor foarte reușite nu mai reprezintă un fel de muncă la jumătate de normă sau o muncă pe care cineva o face pe cont propriu. Acum este era echipelor de proiectare profesioniste și a adevăratelor pagini dinamice, care sunt sensibile la cea mai ușoară trecere a cursorului de mouse peste aproape orice element și necesită cunoștințe de programare pentru a le crea și întreține. Rulează pe platformele Win95/NT, Macintosh și UNIX.

Netscape Communicator 4.0

`http://www.netscape.com/`

Browserul Netscape Navigator, împreună cu facilitățile de e-mail, conferință, agendă și alte unelte utile în afaceri, a creat un set de aplicații pentru rețele, care acoperă multe funcții necesare pentru susținerea lucrului în grup. Communicator/Navigator 4.0 adaugă posibilitatea de navigare pe Internet cu ajutorul tastaturii și alte inovații în stilul limbajului HTML 4.0, pentru a rămâne un concurent puternic, dar ușor avantajat, pentru Microsoft. Cel mai nou browser al său, la fel ca și cea mai nouă versiune de Internet Explorer, împinge limitele multimedia și ale interactivității pe Web, venind cu o gamă amețitoare de facilități disponibile pentru domeniile audio, video, ferestre de tip pop-up, grafică derulantă și alte facilități avansate, pentru a mări importanța siturilor de vârf. Rulează pe platformele Win95/NT, Macintosh și UNIX.

NCSA Mosaic 3.0

`http://www.ncsa.uiuc.edu/SDG/Software/Mosaic/`

Unul dintre browserele de început, încă funcționând bine după toți acești ani, Mosaic este mama tuturor (sau aproape) browserelor grafice, într-un sens foarte ad litteram. Atât Netscape Navigator, cât și Internet Explorer s-au bazat la început pe codul și funcționalitatea browserului Mosaic și încă moștenesc anumite asemănări mai vechi. NCSA (National Center for Supercomputing Applications) Mosaic nu mai este direct susținut, dar rămâne un test bun referitor la ce poate să facă un browser obișnuit. Codul sursă este disponibil, cu unele restricții din cauza aranjamentelor de licență cu Spyglass, iar sursa UNIX X Window este disponibilă gratis pentru uzul personal necomercial. Rulează pe platformele Windows, Macintosh și UNIX X Window.

Opera 3.0

`http://www.operasoftware.com/`

Un browser compact și inovator din Norvegia, cu mulți susținători pasionați, browserul Opera se află într-o continuă dezvoltare. Este unul dintre puținele browsere moderne care

rulează cu succes pe un 386 cu 8M de memorie RAM și un hard-disc de 1.5M! Ferestre multiple, un bun control din partea utilizatorului pentru dimensiunea și machetarea interfeței cu utilizatorul (incluzând reacția acustică pentru multe evenimente), navigare cu ajutorul tastaturii și chiar facilitatea de mărire a paginii (până la de 10 ori), fac din acest browser o alegere bună pentru utilizatorii care au resurse limtate. Puteți face zoom invers pentru a crea un câmp care să conțină miniaturile unei serii de pagini sau puteți cere browserului să încarce toți „copiii" unei pagini, făcând din Opera o alegere bună pentru navigarea comparativă a unui sit. Este foarte rapid și sensibil la comenzi, în special pe calculatoarele rapide. Cea mai nouă versiune are suport pentru JavaScript și alte îmbunătățiri. Rulează pe platformele Windows/Win95.

Tango Browser de la Alis Technologies

`http://www.alis.com/`

Browserul Tango pare un browser imposibil atunci când îl folosiți prima oară pentru a accesa multiplele situri internaționale care afișează „supa alfabetului" pe ecran când urmați o legătură. „Amestecătura" de caractere extinse este minunat transformată în rînduri îngrijite cu caractere perfect aranjate, indiferent de limba folosită. În cazul paginilor care nu își identifică limba, un simplu procedeu de triere o restrânge până ce „supa" se transformă în text. Pentru studenții străini sau creatorii de pagini Web, nu poate exista un browser mai versatil decât acesta. El poate trata cu ușurință scripturi scrise în limba arabă sau devangari, caractere chinezești și japoneze, silabe Kanji, complexul sistem de scriere coreeană, cât și o mulțime de alte scripturi. Are incorporat un program de poștă, iar un program complementar, Tango Creator, vă permite să creați un sit Web într-o limbă străină, fără a fi necesară o versiune specială a sistemului de operare. Lecțiile de limbă nu sunt incluse. Rulează pe platformele Windows.

Lynx

`http://www.crl.com/-subir/lynx.html http://lynx.browser.org http://sol.slcc.edu/lynx/`

Larg folosit de studenți și „contabili", browserul Lynx este un browser doar în modul text, disponibil în aproape orice sistem UNIX. Este pe larg folosit de navigatorii Web cu deficiențe de vedere sau motorii, datorită interfeței cu tastatura ușor de folosit și capacității sale de a fi integrat cu aparatele externe de tip Braille. Codurile sursă sunt disponibile din mai multe surse. Rulează pe aproape toate platformele UNIX și este portabil pe Windows, Win95, Amiga, Macintosh și altele.

Emacs (modul W3) versiunea 4.0

`http://www.cs.indiana.edu/elisp/w3/docs.html`

Un alt browser din lumea UNIX, Emacs W3 (GNUscape) este un instrument extrem de puternic pe care îl puteți explora ani întregi fără ai descoperi limitele și fără a-l părăsi, poate doar în cazul în care vă deconectați. Utilizatorii pe termen lung au tendința să fie atât devotați acestui browser, cât și adepții lui. Acceptă atât modul text, cât și grafic și a fost utilizat pentru a dezvolta multe unelte pentru accesibilitate, inclusiv browserele audio. Documentația conține instrucțiuni explicite despre cum se folosește Emacs W3 ca browser audio. Accesul direct la un limbaj puternic de programare, Emacs Lisp, cât și puterea proprie a sistemului UNIX, fac acest browser capabil, teoretic, de a face aproape orice doriți, dacă aveți timpul necesar proiectării. Codul sursă este disponibil. Rulează pe platformele UNIX, X Window, Win95, Windows și Macintosh. Suport limitat pentru Windows și Macintosh.

AMSD Ariadna (versinea Beta)

`http://www.kd.qd.se/iii/amsd/Ariadna/ariadna.htm`

Browser în limba rusă, complet, ieftin, disponibil pentru platformele Windows și Russian Win95. L-am inclus pentru că mi-a păcut numele lui, deși este un exemplu de browser de care nu a auzit nimeni, dar este vital pentru o comunitate mică de utilizatori. Astfel de browsere specializate există pentru limbile arabă, chineză și alte limbi ne-europene, unele dintre ele având sisteme de scriere care diferă de cel în care citiți acest text. Rulează pe platforme Win95/NT.

Grail 0.3

`http://monty.cnri.reston.va.us/grail/`

Este un browser Web extensibil, scris în limbajul Python, un limbaj de programare orientat pe obiecte și care necesită în principal suport Tcl/Tk pe platformele UNIX. Suport complet pentru HTML 2.0 și în plus o mare parte din 3.2. Este util pentru utilizatorul care dorește să aibă un control complet al platformei browserului, inclusiv capacitatea de a corecta erorile pe măsură ce apar. Nu este recomandat celor slabi de inimă sau celor intimidați de tehnică. Codul sursă este disponibil. Rulează pe platformele UNIX, fiind portabil pe Windows și Macintosh.

HotJava 1.0

`http://java.sun.com/products/hotjava/`

Este un browser extensibil, independent de platformă, de la Sun Microsystems, acceptând toate noutățile din Java, inclusiv JavaBeans. Puteți proiecta propria dumneavoastră interfață grafică cu utilizatorul, folosind limbajul HTML, crea propriile aplicații Web, descărca și executa miniaplicații Java, chiar și prin sisteme firewall și adăuga noi protocoale multimedia oricând doriți. Dacă vă place Java, veți îndrăgi și HotJava. Rulează pe platforme UNIX, Windows și Macintosh.

Amaya 1.1

`http://www.w3.org/Amaya/`

Este bancul de probă de la W3C pentru noi caracteristici și funcționalități. De obicei acceptă cele mai noi standarde și propuneri, dar există o mică întârziere. Codul sursă este disponibil. Este o combinație de editor WYSIWYG și browser, disponibil pentru platformele UNIX și Windows.

Cyberdog 2.0

`http://cyberdog.apple.com/`

Este un browser Web integrat produs de Apple, care include acces la e-mail, filme QTVR Telnet și kituri de instrumente Apple specializate, inclusiv OpenDoc și Bosco – un proiec Web pentru copii. De asemenea, oferă un sistem de răsplătire a celor care fac trimitere la e precum această notă care v-ar putea încuraja să atrageți prietenii spre Apple/Cyberdog Rulează pe platforma Macintosh.

Cello 1.01a

`http://www.law.cornell.edu/cello/cellotop.html`

Nu mai este supus unei dezvoltări active, dar are o comunitate mică de utilizatori, inclusi utilizatori din domeniul academic și alți utilizatori cu acces limitat la un hardware foart puternic necesar browserelor moderne. Cello va rula pe un 386, dar un procesor ma puternic îi va crește performanțele. Ca nu cumva să se simtă superioare unui softwar

„antic" din 1994, majoritatea browserelor moderne vor descoperi că nu pot afișa un fișier Windows de tip bitmap de pe situl Cello, pe când Cello poate să-l afișeze destul de bine. Rulează pe platformele Windows.

Editoare HTML

Editorul HTML reprezintă instrumentul de bază al creatorului de pagini Web. Cele mai bune editoare vă pot ușura munca, fie ascunzând detalii din cod, fie ajutându-vă să-i controlați complexitatea cu verificarea de sintaxă și alte „ajutoare" inteligente. Tabelul G.2 vă prezintă editoarele ce vor fi descrise mai pe larg în această anexă.

Tabelul G.2 Editoare HTML.

Produs	*Tip/Caracteristici*	*Audiența*	*Nivelul de preț*	*Platforme*
Netscape Composer	Semi WYSIWYG	Începător	Gratis, până la $$	Win95 Mac, UNIX
HoTMetaL Pro	WYSIWYG bazat pe cod			Win95/NT UNIX* Mac*
Hot Dog Express	WYSIWYG	Începător	$$	Win95
HotDog Pro	Asistență Expert Administrator de sit Bazat pe cod	Intermediar până la expert	$$$	Win95
Amaya	WYSIWYG	Intermediar	Gratis	UNIX Win95/NT
BBEdit	Bazat pe cod	Începător până la expert	$$	Mac
PageMill	WYSIWYG	Începător până la intermediar	$$$	Mac Win95/NT
HomeSite	Bazat pe cod	Intermediar	$$	Win95/NT
HomePage	Semi WYSIWYG	Începător	$$	Windows
WebEdit Pro	Bazat pe cod	Intermediar	$$	Win95/NT
Webber 32	Bazat pe cod Moduri HTML	Intermediar	$$	Win95/NT
Vis. SlickEdit	Bazat pe cod Moduri HTML Interfață extensibilă pentru managementul de proiect	Intermediar până la expert	$$$	Win95/NT UNIX
Multi-Edit	Bazat pe cod Moduri HTML Manager extensibil de sit	Intermediar	$$$	Win95/NT

Tabelul G.2 Editoare HTML.

Produs	*Tip/Caracteristici*	*Audiența*	*Nivelul de preț*	*Platforme*
	Interfață pentru management de proiect			
Emacs	Bazat pe cod Moduri HTML Extensibil	Expert	Gratis sau ??	UNIX Win95/NT Majoritatea
Vi	Bazat pe cod Moduri HTML Extensibil	Expert	Gratis sau ??	UNIX Win95/NT Majoritatea

Începător: Control slab asupra codului/aspectului HTML, dar relativ ușor de folosit.

Intermediar: Control bun asupra codului/aspectului HTML, dar necesită o anumită îndemnare și experiență pentru a-l folosi.

Expert: Control excelent asupra codului/aspectului HTML, dar necesită o bună îndemnare sau experiență pentru a-l folosi.

OBSERVAȚIE

Multe editoare pot fi încadrate clar, sau cel puțin argumentat, în câteva categorii. În toate tabelele succinte, sunt inevitabile unele suprapuneri și delimitări neclare.

Netscape Composer 4.0

`http://www.netscape.com`

Este un editor semi WYSIWYG, făcând parte din pachetul Netscape Communicator, care vă permite includerea anumitor etichete în pagină, simplu, de la tastatură, într-un element etichetă special. Este ușor de folosit dar limitat ca domeniu de aplicare. Este bun pentru a publica o pagină pe Web, dar este serios depășit când trebuie să facă o lucrare serioasă.

HoTMetaL Pro 4.0

`http://www.softquad.com/`

Acest editor este mama tuturor editoarelor HTML, acum într-o formă nouă care include o mulțime de surprize pentru aproape toată lumea, de la suport intern pentru mediul Microsoft și Dynamic HTML în stil Netscape, până la îndrumare de accesibilitate care vă ajută să vă asigurați că paginile dumneavoastră sunt ușor de folosit de browserele Braille, text și audio. Conține tehnologie AdaptAble, incluzând Visual Dynamic Keyboard pentru a permite utilizatorilor cu dexteritate limitată să acceseze toate caratteristicile programului, asigurând o interfață pentru acești utilizatori pentru a proiecta și scrie codul pentru propriile pagini Web. Acesta este un lucru foarte bun, iar HoTMetaL Pro este foarte recomandat în atenția unei piețe pe care aproape toți ceilalți o ignoră.

HoTMetaL Pro conține facilități de management și vedere de ansamblu a sitului, sintaxă și verificarea legăturii, editare WYSIWYG, editare de pictograme, editare de cod primar și conversie automată din principalele procesoare de cuvinte și programe de calcul tabelar, cât și toate celelalte facilități la care vă puteți gândi când creați pagini Web. Este o alegere

bună atât pentru începători, cât și pentru profesioniști. Nu trebuie confundat cu HoTMetal, o versiune mai veche care se distribuia gratis și care era substanțial mai slabă. Cea mai nouă versiune este pentru platforma Win95 (până acum), dar este acceptat și pe versiuni mai vechi de Mac și UNIX.

Hot Dog Express

`http://www.sausage.com/`

Proiectat pentru creatorii începători de pagini Web, HDE oferă o interfață de tip drag-and-drop (WYSIWYG) simplă, pentru a vă permite să proiectați rapid o pagină gazdă sau pentru alte operații. Poate edita în limbaj HTML direct, dar aceasta este o modalitate rapidă și neplăcută de a aranja o machetare sau un aspect, lăsând alte operații mai complexe pentru alte editoare mai puternice. Rulează pe platformele Win95.

Hot Dog Pro 4.0

`http://www.sausage.com/`

Marele „câine" (dog), conține un kit cu instrumente „supertoolz" profesionale de dezvoltare pe Web, (hei, nu putem fi responsabili pentru „umorul" australian, dar cel puțin ultima versiune a renunțat la ecranul de început cu câinele care urla, în favoarea unui cățel verde care clipește) pentru a crea orice, de la RealAudio/Video cu ajutorul fișierelor `.wav` sau `.avi`, până la un mic bagaj de miniaplicații Java.

Culorile etichetelor, la nivelul limbajului HTML, extensiile Internet Explorer și Netscape Navigator, verificarea sintaxei și a legăturii, etichetele extensibile și alte surprize, fac din HotDogPro o bună alegere pentru o proiectare serioasă. O facilitate de actualizare automată permite actualizarea la ultimele versiuni, ale necesităților programului, dar codul este furnizat cu licență pentru fiecare calculator.

Administrarea sitului, macrocomenzi inline, încărcare automată a unui sit încheiat, împreună cu alte caracteristici, fac din administrarea siturilor Web mari, o operațiune ușoară. Face o editare bazată pe cod. Rulează pe platforma Win95.

Amaya

`http://www.w3.org/Amaya/`

Editor HTML WYSIWYG, creat de W3C. Consultați scurta prezentare a browserului Amaya din secțiunea anterioară despre browsere.

BBEdit 4.5

`http://www.barebones.com/`

Este un editor HTML bazat pe cod, structurat într-un format ușor de folosit. Este multilingvistic, codifică culorile pentru etichete, acceptă etichete extensibile, conține verificare și alte caracteristici care îl fac o alegere excelentă pentru utilizatorii care folosesc platformele Mac și pentru programatorii avansați în HTML.

Adobe PageMill 2.0

`http://www.adobe.com/internet/`

Este un editor comercial care automatizează procesul de găzduire a unui sit care să arate bine, cu cât mai puțin efort. O interfață pas cu pas de tip drag-and-drop (WYSIWYG) facilitează crearea unui sit Web cu o calitate de „muzeu", foarte rapid, chiar și pentru începători. Pentru lucrul direct cu elementele grafice, este inclus programul Photoshop LE.

Este un bun început pentru utilizatorii deja devotați mediului Macintosh, dar nu este la fel de puternic ca mediile bazate pe cod pentru utilizatorii avansați. Cea mai nouă versiune pentru Macintosh conține Site Mill, despre care se va discuta mai târziu în subcapitolul „Publicarea pe Web și administratori de sit". Rulează pe platformele Win95/NT.

HomeSite 2.5 de la Allaire

`http://www.allaire.com/`

Un editor simplu dar puternic, bazat pe cod, realizat de creatorii administratorului de sit „Cold Fusion". Rulează pe platformele Win95/NT.

Claris Home Page 3.0

`http://www.claris.com/products/clarispage/`

Este un editor semi WYSIWYG pentru începători, conținând destule instrumente, inclusiv miniaplicații Java, controale multimedia și alte caracteristici pe care le puteți folosi pentru a vă situa paginile în rândul celor mai bune. Numele spune tot, deci este folosit în principal pentru uz casnic, nu pentru o lucrare Web serioasă. Rulează pe platformele Windows.

WebEdit Pro 2.03

`http://www.luckman.com`

Este un editor bazat pe cod pentru platformele Win95/NT.

Webber 32 de la ExperTelligence

`http://www.webbase.com/`

Este un editor bazat pe cod care conține un suport excelent pentru restricționarea încorporată a etichetelor și atributelor la standarde predefinite, fie propuneri W3C, fie seturi de etichete extinse particulare. Rulează pe platformele Win95/NT.

Visual SlickEdit 2.0 de la MicroEdge

`http://www.slickedit.com/`

Este un adevărat editor pentru programatori și unul dintre cele pe care le folosesc personal (cu adăugarea de extensii și bare de instrumente create de mine), care acceptă o adevărată proiectare bazată pe cod. Are încorporat suport pentru majoritatea limbajelor de programare principale, inclusiv acces direct la API Apprentice pentru rezolvarea apelărilor API dificile, verificatoare de sintaxă și/sau compilatoare, cât și alte caracteristici „de vis" pentru programare. Printre limbajele acceptate sunt Java, JavaScript și Perl, așa că devine evidentă aplicabilitatea acestui editor pe Web. De asemenea, este încorporat și suportul pentru HTML, cu acces direct la browserul Web favorit pentru a vedea cum „progresează" codul dumneavoastră. În viitor vor fi implementate și, în momentul în care citiți aceste rânduri deja vor fi funcționale, îmbunătățiri ale setului de caracteristici HTML, inclusiv codificarea completă a culorilor și verificarea gramaticală inteligentă.

Codul nou scris poate fi marcat cu o culoare diferită, așa că sunt ușor de urmărit liniile de cod modificate. Un set de comenzi foarte extinse împreună cu un limbaj de programare orientat pe obiecte, asemănător cu C++, încorporat, permit ca acest editor să fie ușor de personalizat pentru orice scop, pentru utilizatorii cu experiență în programare. El emulează câteva editoare folosite uzual, cum ar fi Emacs, Vi și Brief, cât și propriul său mod CUA. De asemenea acest editor recunoaște „expresiile regulate".

Viteza deosebită față de editoarele grafice, fac din SlickEdit o alegere perfectă pentru utilizatorii care doresc să facă o lucrare serioasă fără să mai aștepte ca ecranul să termine

actualizarea. Interfața cu sistemele de administrare a proiectului și sistemele de control al codului sursă, inclusiv editarea compusă și diferențiată pentru a facilita urmărirea modificărilor simultane de cod a mai multor programatori pe aceeași sursă și menținerea unui istoric al codului, îl recomandă de asemenea, ca fiind o bună alegere pentru companiile mari dedicate codurilor și formatelor de cod standard, deși nu sunt incluse instrumente specifice de administrare a sitului Web. Versiunea dezvoltată pentru modul text este disponibilă pentru platformele DOS. Rulează pe platforme Windows, Win95/NT, OS/2, UNIX și X Windows. Nu există suport pentru platfoma Macintosh.

American Cybernetics Multi-Edit

`http://www.amcyber.com/`

Este un alt editor de programare adevărat, având un set de caracteristici și o gamă de aplicabilitate pe Web similare cu ale altor editoare, dar cu o componentă opțională WebLair de administrare a sitului, care adaugă o forță considerabilă la dezvoltarea completă a sitului Web. Modurile speciale de programare HTML, administrarea opțională a sitului, editare de atribute sensitivă la context și forța deplină a unui editor care cunoaște totul despre Java, JavaScript, Perl, VBScript, HTML și alte limbaje de programare importante îl fac un concurent puternic pentru Visual SlickEdit, deși nu este disponibil pe tot atâtea platforme ca el, cât și egalul oricărui editor bazat pe cod de pe piața actuală. Emulează editorul Brief, dar nu și Emacs sau Vi. Versiunea pentru modul text este disponibilă doar sub DOS. Rulează doar pe platformele Win95/NT.

Emacs și Vi

`http://www.gnu.org/`

Bineînțeles că editoarele cele mai complete sunt editoarele Emacs și, poate pe o mai mică extindere, editoarele Vi. Deși nici unul nu are acces direct la validarea HTML și la alte caracteristici specifice editoarelor grafice dedicate, cunoașterea aprofundată a unuia sau poate a ambelor editoare este destul de răspândită în lumea programatorilor și există (sau au fost create) pachete de utilizator (*moduri* în Emacs) pentru a mări viteza oricărei operații de programare, inclusiv validarea.

Remarcați că nu am folosit cuvintele „simplificarea operațiilor de programare"; ambele programe sunt extrem de complexe și oferă un sprijin redus sau deloc pentru începători și au susținători (fanatici?) care disprețuiesc editoarele mai slabe. Să privești un guru expert în Emacs sau VI cum scrie cod este ca și cum ai încerca să urmărești mâinile unui magician – puteți fi ușor păcălit, dar trucul vă umple de admirație. Prețul pe care îl plătiți pentru a atinge această îndemânare este o lungă periodă de învățare. Au fost create versiuni pentru aproape fiecare platformă principală și sunt disponibile atât versiunea gratuită, cât și cea comercială. Pe calculatoarele UNIX, ambele editoare sunt prezente în mod implicit sau pot fi obținute de pe siturile Web de la Free Software Foundation (GNU) listate anterior. Rulează pe platformele Win95/NT, Windows, DOS, Macintosh, UNIX, VMS și altele.

Instrumente pentru situri Web : Publicarea pe Web și administratori de sit

La următorul nivel după editor se află programul de publicare sau administratorul de sit, deși există asemănări considerabile. Unele dintre editoarele menționate în capitolul anterior pot fi considerate programe de publicare sau administratori de sit și vor fi deci trecute și în tabelul G.3.

Tabelul G.3 Programe de publicare pe Web și administratori de sit.

Produs	*Tip/Caracteristici*	*Audiență*	*Nivelul de preț*	*Platforme*
HotDog Pro 4.0	Bazat pe cod Administrator de sit Asistență expert	Intermediar până la expert	$$$	Win95
NetObjects Fusion	WYSIWYG Administrator de sit	Intermediar până la expert	$$$	Win95/NT Mac
SiteMill	WYSIWYG Administrator de sit	Începător până la intermediar	$$$	Mac
HoTMetaL Pro	WYSIWYG Bazat pe cod	Novice până la expert	$$$	Win95/NT UNIX Mac
Cold Fusion	Bazat pe etichete Pe partea de server	Intermediar până la expert	$$$	Win95/NT
WebberActive	Administrator de sit			Win95/NT
FrontPage 98	WYSIWYG Asistență expert Administrator de sit	Novice până la intermediar	$$$	Win95/NT

Hot Dog Pro 4.0

`http://www.sausage.com/`

Vedeți prezentarea de la capitolul anterior, „Editoare HTML".

NetObjects Fusion 2.0

`http://www.netobjects.com/`

Este un editor și un program de publicare WYSIWYG pentru care unii se jură, iar alții îl înjură. Avantajele sale sunt date de o bună interfață de tip drag-and-drop, un set de caracteristici moderne și unelte puternice de administrare a sitului, permițându-i să centralizeze operațiunile de luare a deciziilor și controlul asupra proiectelor mari sau echipelor de proiectare distribuite și, de asemenea, îl fac destul de simplu pentru a permite unui începător să configureze un sit Web bine creat fără a fi necesar să învețe prea mult și fără problemele inerente încă de la prima pagină.

Instrumente uimitoare pentru arhitectura sitului permit o concepție simplă, dumneavoastră doar plasați paginile peste o schemă generală, iar ele realizează toate legăturile. Conține un control bun al plasării elementelor grafice și textului cu o bună paletă de controale și alte caracteristici interesante pentru creatorii profesioniști de pagini Web. Totuși, codul produs nu poate fi modificat de mână și arată incredibil de „dezordonat", așa că va ofensa puriștii Web și maeștrii HTML cărora le place să-și „murdărească" mâinile.

Adobe SiteMill 2.0

`http://www.adobe.com/internet/`

Program pentru administrarea sitului, simplu, ușor de folosit pentru utilizatorii cu aptitudini artistice, fiind produsul complementar al programului Adobe PageMill. Când mutați o pagină, Adobe SiteMill reface automat legăturile întrerupte și face toată munca de organizare în locul dumneavoastră, cu un efort mic sau chiar deloc în afara plasării pictogramelor cu ajutorul mouse-ului. De asemenea verifică legăturile off-line și încarcă un sit întreg pritr-o simplă execuție a unui clic, așa că se potrivește bine cu filozofia Apple Macintosh. Rulează doar pe platformele Macintosh, așa că dacă îl primiți împreună cu pachetul Windows, nu prea vă e de folos.

HoTMetaL Pro 4.0

`http://www.softquad.com/`

A fost prezentat la capitolul „Editoare HTML".

Cold Fusion 3.0

`http://www.allaire.com/`

Ați visat vreodată că există o etichetă care să facă ...? Programul Cold Fusion nu se încadrează ușor într-o anumită categorie, dar este un mediu dedicat pentru a simplifica și mări viteza de dezvoltare a unui sit Web, „extinzând" limbajul HTML cu etichete speciale care nu aparțin limbajului HTML și care pot accesa baze de date „direct". Are facilități de email, procesare a cărților de credit și realizează aproape tot ceea ce doresc utilizatorii HTML fanatici. Face parte dintr-un pachet, care necesită un Cold Fusion Application Server (doar pe platformele NT) pentru a funcționa cu orice server standard HTTP și a realiza operațiile dinamice care îl fac atât de puternic.

Există puține lucruri care nu pot fi făcute cu ajutorul scripturilor CGI de utilizator, dar cine are timp să scrie tot codul necesar să acopere teritoriul Col Fusion? Și pe deasupra, cine are timp să învețe toate aplicațiile API cerute pentru a folosi ActiveX sau ODBC la toată capacitatea lor? Dacă interfețele incluse nu sunt destule, puteți chiar scrie propriile dumneavoastră interfețe pentru etichete pentru funcționalitatea dorită, informând Application Server despre ele. Cold Fusion 3.0 este un browser Web dosebit care pare să înțeleagă etichetele `<PIRATE>` și `<PINHEAD>`, deși trebuie să le denumiți `<CF_PIRATE>` și `<CF_PINHEAD>`. Bineînțeles nu prea le înțelege, dar aceasta nu se mai spune. Rulează pe platformele Win95/NT.

WebberActive 1.1 de la ExperTelligence

`http://www.webbase.com/`

Este un utilitar pentru editare și administrarea sitului, care include și suport pentru Dynamic HTML și Active Channels. Este foarte apropiat de standardele și browserele Microsoft, deși versiunea curentă încă acceptă etichetele limbajului HTML 3.2 inițiale. Rulează pe platformele Win95/NT.

Microsoft FrontPage 98

`http://www.microsoft.com/sitebuilder/`
`http://www.microsoft.com/products/developer.htm`

Este un alt editor/administrator de sit de tip WYSIWYG cu programe wizard care vă asistă la crearea paginilor Web. Având o mulțime de facilități Microsoft, ocupă mult spațiu pe hard-disc dar conține multe controale grozave, cum ar fi Active Channels și alte surprize create de Microsoft, cum ar fi programe wizard de baze de date pentru a putea prelua

conținutul altor baze de date. Este foarte bine integrat cu produsele Microsoft, dar modelele cu care vine sunt puțin cam lipsite de imaginație. Pe de altă parte, ele sunt mult mai bune decât jumătate din paginile pe care le vedeți pe Internet, așa că este o alegere excelentă pentru începători sau pentru acei utilizatori care încă nu și-au creat un stil propriu. Are instrumente bune pentru administrarea sitului, dar nu la fel de bune ca editorul NetObjects Fusion. Rulează pe platformele Win95 și Macintosh.

Instrumente pentru situri Web: Hărți de situri

Majoritatea administratorilor de sit au disponiblie hărți de situri bune, dar uneori doriți ceva care să fie de sine stătător, fie deoarece administratorul de sit nu este destul de flexibil, fie deoarece doriți ceva care se autoactualizează, ca să nu trebuiască să actualizați harta de câte ori se modifică situl. Există două modele de bază pentru maparea sitului: un model ierarhic, care se mapează bine într-un director de tip arbore și o reprezentare grafică neierarhică, ce reflectă mai bine câte situri sunt organizate și chiar cum este organizată rețeaua Web însăși. Viitorul pare să fie al instrumentelor grafice multidimensionale, deoarece utilizatorii nu pot înțelege sau naviga decât printr-un director de tip arbore simplu.

Realitățile virtuale „cyberspace" atât de frumos descrise de Peter Gibson și alții sunt toate mapatoare de sit, extraordinar de detaliate și interactive, ce creează o interfață, folosind magia tehnologiilor viitorului, direct spre creier. Aceasta pare o fantezie specifică bărbaților, din punctul meu de vedere; nu ar fi mai ușor să se oprească și să întrebe care sunt indicațiile?

Bineînțeles, motoarele principale de căutare, ar putea fi considerate ca instrumente de mapare a sitului, dar ele sunt cunoscute destul pentru a nu mai discuta despre ele. Totuși, unele pot fi instalate pe un sit local, asigurând un anumit nivel de funcționalitate pentru căutare/mapare a sitului. Vă prezentăm mai jos câțiva candidați mai specializați, inclusiv câțiva care pot fi incluși în paginile dumneavoastră la prețul unei legături la situl proiectantului. Mare afacere! Tabelul G.4 vă oferă o comparație rapidă a instrumentelor de mapare a sitului.

Tabelul G.4. Unelte de mapare a sitului.

Produs	*Tip/Caracteristici*	*Audiență*	*Gama de preț*	*Platforme*
MapXsite	Geografic	Expert	$$$	WinNT
SiteAnalyst	Hartă de sit	Intermediar	$$$	Win95/NT
CLEARWeb	Hartă de sit	Oricine	$$	Win95/NT
MAPA	Service de hartă	N/A	$$$$	Toate
WebAnalyser	Hartă de sit	Oricine	$$	Win95/NT
SiteCommand	Statistică Hartă de sit	Intermediar	$$$	WinNT
HotSauce	Statistică Hartă de sit	Oricine	$$$	Mac Win95/NT
WebCutter	Vizionare în 3D Hartă de sit	Oricine	???	Toate
Visual SiteMap	Vizionare în 3D Hartă de sit	Expert	N/A	UNIX
Web Collection	Articol	Oricine	Gratis	N/A
Site Map	Hartă de sit	Oricine	Gratis	Toate
Web Site Expl.	Hartă de sit	Oricine	Gratis	Toate
Governor	Hartă de sit	Oricine	Gratis	Toate
Exploder	Hartă de sit	Oricine	$$$	Toate

MapXsite de la MapInfo

`http://www.mapxsite.com/ WinNT platform`

Este un produs comercial pentru platformele WinNT, care permite crearea de hărți 2D navigabile, pe care se poate executa clic, se poate face zoom și se pot face căutări locale bazate pe geografie adevărată. Deși nu se aplică specific la maparea unui sit logic, pentru situri cu o metaforă geografică puternică, este un instrument foarte puternic, cu acces incorporat la bazele de date cu informații goegrafice (Geographical Information Systems – GIS) și alte caracteristici care facilitează trecerea de la realitatea fizică la elementele logice. Rulează pe platformele WinNT.

Microsoft SiteAnalyst (Fostul WebMapper de la NetCarta)

`http://backoffice.microsoft.com/products/features/SiteAnalyst/SiteServerE.asp`

Este un instrument 2D (bidimensional) pentru analizarea și întreținerea siturilor Web, fără să asigure o hartă de navigare pentru utilizatori. Face parte din suita de programe BackOffice pentru administratorii de rețea și Webmasteri. Rulează pe platformele Win95/NT.

CLEARWeb

`http://www.clearweb.com/`

Este un instrument pentru maparea sitului, bazat pe Java, care afișează atât structura directoarelor, cât și o hartă grafică. Acest instrument se adresează unor Webmasteri și utilizatori care doresc să-și organizeze propria lor colecție de semne de carte și referințe. Este acceptat de browserele Internet Explorer și Netscape Navigator. Rulează pe platformele Win95/WinNT.

MAPA Service de la Dynamic Diagram

`http://www.dynamicdiagrams.com/`

Este un mapator de sit, pseudo 3D (tridimesional), bazat pe Java care poate aduce pe ecran o vedere la nivel de „hartă stradală" a unui sit și ferestre de tip pop-up pentru a afișa mai multe informații în timp ce investigați situl folosind un mouse. Din punct de vedere vizual este foarte plăcut, dar este încă o vizualizare analogică și o prezentare a unui simplu director cu o structură ierarhică tip arbore. Se vinde doar ca versiune de încercare, așa că platforma pe care rulează nu este relevantă. Compania realizează de asemenea și modelare în 3-D, așa că nu prea este clar dacă există posibilitatea de mapare 3D a sitului, dar se pare că nu există. Hărțile există pe server și sunt vizibile în orice browser care acceptă limbajul Java.

WebAnalyzer 2.0 and Site Command 1.01 de la InContext

`http://www.incontext.com/`

Este un kit de instrumente pentru grafică, bazat pe text, pentru vizualizarea unui sit Web ca o ierarhie de pagini legate sau un front de noduri conectate într-un graf direct, cu posibilitatea de a alege oricare nod ca punct central. Recent a fuzionat cu EveryWare Development Canada, așa că adresa de Web s-ar putea să se fi schimbat. Nucleul SiteCommand (versiunea „Pro" va fi lansată curând) rulează pe un număr limitat de servere Netscape și Microsoft și permite hărți de sit bazate pe text și date statistice extinse pentru a putea fi construite și analizate dinamic de Webmaster, dar nu este necesar în cazul folosirii ocazionale sau intermitente pe un sit mic. WebAnalyzer se adresează, de asemenea, unui Webmaster, oferind multiple vederi ale unui sit, inclusiv afișare grafică, pentru a permite urmărirea legăturilor întrerupte, fișierelor copiate, a fișierelor cu prea mult conținut și a

altor informații cu folosire limitată pentru utilizatori, deși hărțile în sine ar putea fi folositoare. Rulează pe platforme Win95/NT.

HotSauce de la Apple

`http://hotsauce.apple.com/`

Este un browser grafic 3D de tip plug-in, care necesită scrierea unui fișier MCF (Meta Content Framework) pentru a descrie modelul interactiv în 3D pe care navighează HotSauce. Are o concepție frumoasă, dar dă mult de lucru celor mai mulți Webmasteri și, bineînțeles, nu prea puteți obține o proiecție prea bună fără un plug-in extern. Totuși, face parte din ofertele altora, inclusiv NetObjectsFusion, care poate genera pe loc o descriere MFC și câteva alte sisteme. Pentru un mediu intern sau alt auditoriu restrâns, are posibilități adevărate ca o rezumare de utilizator, dar ca instrument de uz general care să atragă vizitatorii pe situl dumneavoastră, mai lasă de dorit deoarece necesită un plug-in. Rulează pe platformele Mac și Win95/NT.

WebCutter (Lotus Domino)

`http://www-ee.technion.ac.il/W3C/abstract.html`
`http://www-ee.technion.ac.il/W3C/WebCutter.html`

Este un instrument bazat pe Java, care afișează dinamic hărți de sit, atât în stilul director (arbore), cât și în stilul geografic-logic (grafuri direcționate sau rețele), bazate pe fișiere text ASCII, create de un Webcrawler special, care descrie situl. Maparea contextuală furnizează o pseudo imagine în 3D a sitului, ca o colecție de grafuri ale căror noduri sunt grupate prin egalizarea distanței vizuale și a semnificației logice a nodului ales ca punct de pornire. Aceasta este o caracteristică foarte bună și ar putea eventual deveni parte integrantă a pachetului Lotus Notes/Domino. Rulează pe orice platformă Java.

Visual Site Map

`http://lislin.gws.uky.edu/Sitemap/Sitemap.html`

Este prototipul hărții Visual Site Map, realizat de Universitatea din Kentucky. Această hartă este un instrument offline bazat pe o rețea neurală, pentru analizarea și prezentarea relațiilor logice între anumite porțiuni dintr-un sit, ținând cont mai mult de conținut decât doar de legături. Este un proiect experimental, dar interesant, deoarece probabil reprezintă o direcție viitoare spre care se îndreaptă maparea siturilor. Nu depinde de nici o platformă.

WebCollection (W3C Draft)

`http://www-ee.technion.ac.il/W3C/WebCollection.html`

Un articol de bază despre o parte din lucrările care se fac la W3C referitor la maparea sitului. Nu depinde de o anumită platformă.

Site Map de la Joe Bart

`http://junior.apk.net/~jbarta/`

Este o hartă de sit bidimensională, foarte pusă la punct, în stil Explorer în JavaScript, cu indicații de tip „te afli aici" și bannere de tip schimb de legătură. Harta se dezvăluie într-o fereastră flotantă, așa că o puteți deplasa. Rulează pe orice platformă.

Web Site Explorer de la Rabbit Internet

`http://www.rabbitnet.com/`

Este o altă hartă de sit în 2-D (bidimesional), în stil Explorer scrisă în JavaScript. Rulează pe orice platformă.

Governor Site Map

`http://www.governor.co.uk/`

Este o hartă de sit în stil Explorer 2-D, scrisă în Java, care se distribuie gratis. Foarte rapidă și flexibilă. Rulează pe orice platformă.

Exploder Site Map Tool de la Beach Software

`http://www.halcyon.com/beach/products/applets/`

Acesta ese o altă hartă de sit în 2-D, scrisă în limbajul Java, de asemenea foarte portabilă, deși fundalul este de culoare gri închis dintr-un anumit motiv. Cred că este mult mai impresionant datorită codului decât datorită esteticii,. Este un produs comercial, dar codul sursă scris în limbajul Java, este inclus în prețul de achiziție, așa că puteți să-l prelucrați cum doriți. Rulează pe orice platformă cu suport Java.

Instrumente pentru situri Web: Diverse produse shareware

Există multe unelte bune la un preț redus, ca shareware, prea multe pentru a fi listate aici deoarece apar tot timpul noi programe pe piață. Ele sunt disponibile pentru a fi descărcate de pe câteva situri, dar eu prefer situl TUCOWS deoarece el evaluează programele și vă oferă un scurt sumar. Celelalte două situri menționate în această secțiune (și listate în Tabelul G.5) au de asemenea evaluări și o gamă mai largă de produse.

Tabelul G.5. Diverse produse shareware.

Produs	*Tip-Caracteristică*	*Audiență*	*Gama de preț*	*Platforme*
TUCOWS	WinSock Ware	Oricine	Variază	Win95/NT
ZDNet	Shareware	Oricine	Variază	Mai multe
CINet	Shareware	Oricine	Variază	Mai multe

TUCOWS Central

`http://www.tucows.com/`

Acest sit este specializat în instrumente WinSock, așa că are o acoperire relativ redusă pentru editoarele grafice sau asemănătoare lor. Numele vine de la „The Ultimate Collection Of WinSock Software", deși se pare că producătorii s-au cam jucat cu cuvintele datorită faptului că se folosesc abundent pe sit poze cu vaci (cow = vacă), jocuri de cuvinte proaste despre vaci și alte bancuri cu vaci. Deoarece WinSock apare de obicei cu scris ciudat cu litere mari, cred că ei s-au gândit mai întâi la jocul de cuvinte și apoi au modelat numele ca să aibă sens, așa că au putut folosi poze cu vaci drăguțe pe sit. Rulează pe platformele Windows și Macintosh.

Software Library de la ZDNet

`http://www.hotfiles.com/need description`

Ziff-Davis, editorul revistelor *PC Magazine, PC Week, MacWEEK* și a altor reviste din domeniul calculatoarelor, prezintă cu regularitate recenzii și comparații ale produselor software în paginile revistei sale. De fapt, el are o echipă care testează cu regularitate noile produse punând pe sit recenziile pentru a fi citite de oricine. Revistele sunt de departe cel mai bun (și mai oportun) loc de a găsi recenziile, deoarece ei întârzie publicarea pe Web

pentru a nu influența vânzările propriilor reviste, incluzând de obicei și mai multe informații. Dacă aveți nevoie urgentă de informații și nu aveți timp să citiți revistele, situl Web este o alternativă bună. Sunt valabile toate platformele, dar există un accent pe platformele Intel și Macintosh.

Download.com de la CINet

`http://www.download.com/ditto`

C|Net este cel mai cunoscut producător de emisiuni de informare din domeniul „high-tech" prezentate la TV, incluzând „The New Edge", „C|Net Central", „TV.COM" și „The Web". Ei au și o emisiune radio pe Web și situl de știri pe Web, NEWS.COM. Ca parte integrantă din acest potpuriu de „tehno-conversație", ei oferă Download.com, un sit care prezintă noutățile legate de shareware, demoware și alte informații de senzație pe care să le îngrămădiți într-un colț al hard-discului, unele dintre ele fiind recenzate și evaluate. Rulează pe toate platformele, dar cu accent pe platformele Intel și Macintosh.

BIBL. CENTR. UNIV. „M. EMINESCU" IAȘI INFORMATICĂ

Grafică

Dacă o imagine poate înlocui o mie de cuvinte, de ce benzile desenate au cuvinte? Serios vorbind, o grafică bună poate face ca un sit Web să arate impresionant, dar imaginile bune nu sunt destul fără cuvinte. Aveți nevoie și de imagini și de cuvinte și aveți nevoie de instrumente bune pentru a face ca imaginile să funcționeze bine pe Web.

Există o mulțime de utilitare ieftine și eficiente, disponibile pentru începători (vedeți Tabelul G.6), cum ar fi GIF Construction Set (special pentru animație simplă pe platformele Windows) și LView Pro, ambele disponibile ca shareware pe principalele situri de descărcare. Există și programe de foto-editare cum ar fi Paint Shop Pro, care nu vă vor pune prea multe probleme. Puteți face o mulțime de lucruri cu ele. Mult mai important, puteți învăța.

Dar când aveți nevoie de ceva mai mult, vă gândiți la Adobe Photoshop Illustrator și alte instrumente grafice profesionale. Aceasta nu este din cauză că aceste instrumente sunt extrem de mari, dar ele sunt atât de răspândite în lumea serioasă a proiectării grafice, că multe dintre utilitățile adăugate sunt scrise pentru a lucra cu ele. De asemenea, ele economisesc timpul lucru care după cum știți separă utilizatorii care dorm noaptea de cei care la ora 4 a.m. încearcă să termine o lucrare la timp, în exact șase ore, pentru prezentarea în fața clientului.

Tabelul G.6. Resurse grafice.

Produs	*Tip/Caracteristici*	*Audiență*	*Gama de preț*	*Platforme*
Photoshop	Foto-editor	Profesioniști	$$$$	Win95/NT
Illustrator	Lucrări artistice vectoriale	Profesioniști	$$$$	Mac Win95/NT
	Editor			Mac
Fractal Paint	Foto-editor	Profesioniști	$$$	Win95/NT
				Mac
CorelDRAW!	Lucrare artistice vectoriale	Profesioniști	$$$	Win95/NT
	Editor			
PHOTO-PAINT	Foto-editor	Profesioniști	$$$	Win95/NT
DeBabelizer	Instrument grafic	Profesioniști	$$$	Win95/NT Mac

(continuare)

Tabelul G.6. – continuare

Produs	*Tip-Caracteristici*	*Audiență*	*Gama de preț*	*Platforme*
The GIMP	Foto-editor	Profesioniști	Gratis	UNIX X Window
Freehand 7	Lucrări artistice vectoriale Editor	Profesioniști	$$$	Win95/NT Mac
GIF Construct	Utilitar GIF	Oricine	$$	Win95/NT
Colorworks W3	Foto-editor	Oricine	$$$	Win95/NT
Paint Shop Pro	Foto-editor	Oricine	$$	Win95/NT
LView Pro	Utilitar de imagini Utilitar GIF	Oricine	$$	Win95/NT
GIFTool	Utilitar GIF	Oricine	Gratis	UNIX
GIFTrans	Utilitar de	Oricine	Gratis	UNIX
XV Image View	imagine Utilitar online	Oricine	Gratis	UNIX X Window
Imaging Mach.	Tutorial	Oricine	Gratis	Oricare
Dmitry's Labs	Tutorial	Oricine	Gratis	Oricare
Web Wonk	Tutorial	Oricine	Gratis	Oricare
Photoshop Tips	Tutorial	Oricine	Gratis	Oricare
Homegurrrl	Tutorial	Oricine	Gratis	Oricare
Top Ten Mistakes		Oricine	Gratis	Oricare
Free Icons	Pictograme	Limitat	Gratis	Oricare
Wagon Train	Pictograme	Oricine	Gratis	Oricare
GIF's	Clipart Legături			
	Referințe	Oricine	Gratis	Oricare
Encyclopedia	Clipart	Oricine	Gratis	Oricare
Barry's Clip Art Server				

Adobe Photoshop / Illustrator

`http://www.adobe.com/prodindex/`

Există două tipuri de lucrări artistice folosite pe Web: grafica vectorială și fotografiile (bitmap sau grafice cu rastru). Adobe conține uneltele Cadillac pentru ambele tipuri – Photoshop, care oferă lucrul și finisarea imaginii fotografice și Illustrator, care este destinat graficii vectoriale.

Ambele instrumente au loc pe suprafața de lucru a calculatorului, dar cea mai nouă versiune de Photoshop conține o multitudine de caracteristici noi pentru creatorii profesioniști pe Web, inclusiv suportul pentru PNG, noul format de fișier grafic, care ar putea în cele din urmă să înlocuiască formatele GIF și filigran digital (steganografy), astfel că puteți plasa discret informații de copyright în fiecare imagine, care poate fi recuperată, chiar și după tipărirea și rescanarea imaginii!

Adobe Illustrator adaugă facilități cum ar fi paleta Action, care vă permite să înregistrați pașii necesari creării unui efect și să îi reaplicați în mod automat la mai multe fotografii, că și disponibilitatea unui al treilea pachet, cum ar fi Kai's Power Tools sau Alien Skin. Astfe obțineți o platformă cu adevărat minunată pentru realizarea lucrărilor artistice.

Illustrator este la fel de puternic la crearea graficii vectoriale (acelea pe care le desenați în creion și tuș, spre deosebire de utilizarea unei pensule), iar aceste două instrument lucrează foarte bine împreună deoarece grafica vectorială trebuie în mod uzual să fi convertită într-un format bitmap înainte de publicarea pe Web.

Dacă intenționați să introduceți imaginile dumneavoastră într-un magazin de grafic pentru a le transforma în diapozitive sau să realizați o copiere hard profesională imaginilor, sau chiar să încredințați altora o parte din munca dumneavoastră, majoritate profesioniștilor au doar un software Adobe disponibil și pot accepta doar fișiere în forma Adobe. Rulează pe platforme Macintosh și Win95.

Fractal Design Painter

`http://www.fractal.com/`

Acesta este un alt pachet grafic de tip fotografic, îndreptat spre producerea de efecte d pictor și crearea de lucrări artistice digitale, în locul retușării sau procesării imaginilo scanate. O tablă grafică cu acționare prin apăsare conferă un real avantaj acestui instru ment, așa cum este în majoritatea editoarelor principale de grafică. Painter acceptă creează fișiere Adobe și alte formate de fișier, așa că vă puteți muta ușor de la un format l altul. Acceptă de asemenea Ray Dream Studio pentru procesare în 3-D și animație. Ruleaz pe platformele Macintosh și Win95.

CorelDRAW!/PHOTO-PAINT

`http://www.corel.com/`

Este o bună a doua alegere (unii ar spune că este singura variantă secundară) pentr creatorii profesioniști. Deși suportul nu este atât de acoperitor cum este pentru Adobe, există, așa că nu veți crea un fișier pe care nimeni altcineva nu îl va putea folosi dacă vă ve decide să tipăriți lucrări sau să le executați în cadrul unor comenzi contractuale.

Fișierele COREL au reputația de a fi puțin incomode, dar un avantaj este acela că CorelDRAW PHOTOPAINT este mai puțin costisitor și totuși oferă destulă putere pentru a fi folosito De fapt, mulți proiectanți au instalat ambele programe deoarece fiecare are puterea s specifică. Puteți comuta fișierele între cele două programe pentru a face operațiile m ușoare decât dacă ar fi fost pe unul sau altul.

Ediția curentă vine cu ambele programe într-un singur pachet, cât și cu CorelDREAM 3I un pachet de redare tridimensională și câteva alte lucruri folositoare. Corel accep filigrane, Kai's Power Tools și câteva alte surprize pe care le veți găsi în Photoshop, cât și mulțime de imagini de colecții de clip-art gratuite. Este un instrument foarte bun, așa dacă puteți, faceți rost de ambele. Rulează pe platformele Win95/NT.

DeBabelizer

`http://www.equil.com/`

Un instrument de bază pentru producerea lucrărilor pe Web, permițându-vă să fac prelucrări, prin fișiere de comenzi, pe mai multe fișiere simultan, inclusiv găsirea ur palete optime pentru un set de imagini și impunerea acelei palete asupra tuturor. Put face programul să supravegheze un fișier în timp ce executați o operație complexă și ast

el știe cum să procedeze cu celelalte fișiere în același mod. E ca și când ați avea un „ucenic vrăjitor" care să facă grosul în locul dumneavoastră, dar nu și tot așa de periculos. Un instrument grozav. Rulează pe platforme Macintosh și Win95.

The GIMP

`http://scam.xcf.berkeley.edu/~gimp/`

The GIMP (Gnu Image Manipulation Program) este un editor grafic și un instrument foto de retușare pentru UNIX X Window, cu aceleași caracteristici generale pe care le găsiți în Photoshop, inclusiv filtre, programe plug-in și efecte speciale pentru a realiza transformări complicate ale imaginilor. Ca exemple de browsere care au efecte disponibile, sunt Mozaic, care redă o fotografie ca și cum ar fi mozaicată cu mici bucăți de ceramică; Carven, pentru inserarea unui efect stil Dragnet în 3D într-o fotografie; IFS care creează automat modele de plante, simulând creșterea plantelor; și Chrome, care atribuie imaginii un aspect metalic. Rulează pe platforme UNIX X Window.

Macromedia Freehand Graphics Studio 7

`http://www.macromedia.com/`

Este un instrument puternic pentru crearea de grafică vectorială animată pe Web, instrument bine integrat cu alte produse Macromedia. Rulează pe platformele Win95/NT și Macintosh.

GIF Construction Set by Alchemy Mindworks

`http://www.mindworkshop.com/alchemy/alchemy.html`

Alchemy Mindworks realizează și Graphics Workshop, care vă permite să convertiți multe tipuri de formate grafice dintr-unul în altul, cât și alte trucuri. Ambele pot crea GIF-uri transparente și pot realiza o manipulare simplă a imaginilor, dar doar GIF Construction Set vă permite să creați GIF-uri animate pentru Web, în multe cazuri fiind o alternativă ieftină la ieșirile video și acceptată de aproape toate browserele moderne fără a fi necesare programe plug-in sau programe de vizionare speciale. Rulează pe platformele Windows și Win95.

ColorWorks: Web 3 de la SPG

`http://www.spg-net.com/`

Un nou pachet grafic inovator, care se autointitulează „Mama tuturor suitelor de grafică pe Web", Web 3 oferă o multitudine de facilități pentru Web, incluzând optimizarea animației GIF cu ajutorul căreia puteți micșora automat fișierele animate până la 50% sau mai mult, un utilitar de mapare a imaginii, „reparare de imagine" JPEG, un browser cu bibliotecă de imagini și (aceasta nu o recomand pentru utilizări ocazionale, fără permisiune pe siturile cu copyright) un „păianjen" de grafică Web, care înhață automat toate imaginile dintr-un sit pentru folosirea ulterioară.

Se oferă gratis programe plug-in, pentru a fi descărcate de pe situl SPG și intenționează să ofere suport pentru programele plug-in pentru Adobe Photoshop în viitorul apropiat. Există de asemenea o ofertă de actualizare gratuită pentru 30 de zile de la data achiziționării, pentru utilizatorii care cumpără o versiune care va fi actualizată în curând. Și rețineți – nu trebuie să o căutați dumneavoastră. Ei vă vor trimite mesaje pe e-mail pentru a vă întreba dacă doriți actualizarea și dacă adresa de livrare este corectă. Puteți renunța la actualizări fără a fi penalizați. Iată o responsabilitate socială incredibilă (și neobișnuită) față de clienții fideli ! Nu numai din cauza aceasta, dar unul dintre „mascotele" lor

supererou, cel mai bun proiectant Web al lor, este o femeie. Sper ca Web 3 să o pună pe pagina de deschidere, dar nu trăim într-o lume perfectă. Rulează pe platformele Win95/NT.

Paint Shop Pro

`http://www.jasc.com/`

O alegere bună pentru folosirea ocazională sau un „hobby" pentru proiectarea paginilor Web, Paint Shop Pro este disponibil ca shareware, cu o taxă moderată pentru înregistrare sau poate fi comandat de la cei care l-au creat. Are un număr surprinzător de facilități și acceptă chiar și programe plug-in pentru Adobe Photoshop, așa că o mare parte din forța instrumentelor profesionale vă poate aparține pentru un preț relativ modest. Rulează pe platformele Windows și Win95.

LView Pro

`http://www.lview.com/`

Este un convertor shareware de fișiere în format grafic și manipulator de imagini care include crearea GIF-urilor transparente, animație și editare simplă. Adesea este disponibil pe siturile shareware de descărcare sau îl puteți găsi pe pagina lor gazdă. Rulează pe platformele Windows și Win95/NT.

GIFTool

`http://www.homepages.com/tools`

Este un instrument GIF cu linie de comandă pentru crearea fișierelor GIF întrețesute. Din nefericire, pagina gazdă listată aici a dispărut, poate permanent, dar este încă disponibilă în codul sursă FreeBSD UNIX. Veți avea nevoie de acces la un compilator C (de obicei cc pe majoritatea sistemelor UNIX) pentru a-l compila. Iată adresa unei surse pe care o găsiți în Rusia, dar s-ar putea să găsiți o copie și pe plan local:

`http://www.cronyx.ru/pub/pub/FreeBSD/distfiles/giftool.tar.gz`

Unele versiuni, de asemenea, pot să realizeze transparența, lucru pentru care s-ar putea să aveți nevoie de un instrument de indexare a culorilor, cum ar fi John Bradley's XV Image Viewer (prezentată ulterior în acest capitol). Editorul dumneavoastră grafic probabil are o modalitate de a identifica culoarea fundalului, deși va trebui să convertiți manual în hexa. Totuși, pentru programele cu linie de comandă, poate fi un truc dificil deoarece interfața este doar text și nu veți putea vedea de fapt grafica. Rulează pe platformele UNIX.

GIFTrans de la Andreas Ley

`ftp://ftp.rz.uni-karlsruhe.de/pub/net/www/tools/`

Este un creator de transparență pentru imagini în format GIF, cu linie de comandă. S-ar putea să aveți nevoie de un instrument cum ar fi programul John Bradley's XV Image Viewer pentru a selecta culoarea transparenței, dacă nu o cunoașteți deja, deși aproape orice program grafic vă va permite să vedeți culoarea. Poate va trebui să convertiți dintr-o valoare zecimală RGB într-o valoare hexa, în funcție de programul folosit. Este disponibil codul sursă, cât și un fișier `.exe` sub DOS. Rulează pe platformele UNIX și DOS.

XV Image Viewer de la Joe Bradley

`http://www.trilon.com/xv/`

Dacă folosiți X Window sub UNIX, acest program s-ar putea să existe deja în sistem sau îl puteți obține de pe situl autorului. Conține o mulțime de lucruri cu care să vă distrați

inclusiv obținerea indexului culorii oricărui pixel dintr-o imagine. Sunt acceptate multiple formate de fișiere și se pot realiza conversii. De asemenea, are disponibil pentru descărcare o versiune binară pentru GFITrans. Codul sursă este disponibil, dar nu este gratis. Rulează pe platformele UNIX X Window.

The Imaging Machine

`http://www.vrl.com/Imaging/`

Dacă aveți acces la propria pagină de Web, puteți introduce adresele URL ale fișierelor în acest program și el le convertește în fișiere GIF transparente, întrețesute sau animații de până la 25 de imagini. Puteți obține imaginile înapoi, prin capturarea imaginii de pe ecran. Rulează pe orice platformă, dar aveți nevoie de acces la o pagină Web.

Design Lab de la Dmitry Kirsanov

`http://www.webreference.com/dlab/`

Este un sit oferit de unul dintre colaboratorii acestei cărți, care explorează folosirea fotografiei în crearea paginilor Web și alte chestiuni referitoare la proiectarea grafică. Din această ofertă bine echilibrată fac parte diferite legături numite de autor „dessert links" (legături pentru desert). Nu depinde de anumite platforme.

Web Wonk de la David Siegel

`http://www.dsiegel.com/tips/`

Este o pagină de sfaturi și trucuri de la unul dintre cei mai faimoși creatori de pagini Web. Conține o mulțime de informații de bază, de la tipografie, la ce culoare să dați fundalului pentru a avea cele mai bune imagini și face haz de necaz la încărcare. Merită să fie vizitat atât de începători, cât și de cei versați. Nu depinde de anumite platforme.

Photoshop Tips de la Laurie McCanna

`http://www.mccannas.com/pshop/photosh0.htm`

Întregul sit al lui Laurie, din care această pagină reprezintă doar o mică parte, conține informații vaste referitoare la proiectare, elemente artistice gratuite, liste de legături și alte articole nemaipomenite pentru a fi citite sau copiate. Explorați-l cu încredere. Nu depinde de nici o platformă.

Homegurrrl Page de la Lynda

`http://www.lynda.com/`

Acest sit are multe informații disponibile, inclusiv liste poștale, discuții despre GIF-urile animate și culori, o selecție de legături și alte articole de bună calitate pentru creatorul de pagini Web. Nu depinde o anumită platformă.

Top Ten Mistakes in Web Design

`http://www.useit.com/alertbox/9605.html`

Este eseul incontestabil al lui Jacob Nielsen (Primele zece greșeli în proiectarea Web) referitor la problemele paginilor orfane, culorile nestandard pentru legături și altele. Acesta este doar unul dintre multele articole bune despre proiectarea paginilor Web, grafică, accesibilitate, machetarea paginilor, tipografie și alte articole interesante pentru creatorii de pagini Web, care pot fi accesate toate prin intermediul acestui sit. Nu depinde de nici o platformă.

Free Icons by Goff

`http://members.aol.com/tedgoff/icons.html`

Conține câteva pictograme noi, de dimensiuni mari, create de un realizator de benzi desenate profesionist, ale cărui desene au fost publicate în *„The Saturday Evening Post"*, *„The Wall Street Journal"*, *„Better Homes & Gardens"*, *„Good Housekeeping"*, *„Air & Space"* și *„Harvard Magazine"*. Dacă ați tânjit după o pioctogramă a nasului sau o pictogramă cu o față verde a unui om care pare să fie bolnav, aici le veți găsi. Rulează pe platformele UNIX, Macintosh și Win95/NT.

Wagon Train Animated GIF

`http://dreamartists.com/animated.htm`

Situl încântător și exuberant al lui Carol Waggoner conține un bagaj de lucrări de artă originale pe care le puteți folosi pe situl dumneavoastră sau doar admira. Aceasta este adevărata Americana, primitoare și aglomerată ca un magazin vechi de provincie, o muncă în care a pus suflet și o prezentare a lucrărilor ei. Conține de asemenea o mulțime de legături spre alte situri unde puteți găsi mai multe imagini clip-art, spre deosebire de siturile mai puțin generoase cu lucrări de artă „gratuite" pe care le găsiți pe Web. Rulează pe platformele UNIX, Macintosh și Win95/NT.

Encyclopedia of Graphics File Formats

`http://www.ipahome.com/gff/textonly/book.htm`

Este un ghid indispensabil pentru creatorii profesioniști de pagini Web. Acoperă în detaliu fiecare aspect al formatelor GIF, JPEG, PNG și alte formate grafice. Există o mulțime de alte resurse grafice disponibile pe situl lui Bob MacDonald, așa că merită să-l cercetați. Nu depinde de anumite platforme.

Clip Art Server de la Barry

`http://www.barrysclipart.com/mdex.html`

Există atât de multe situri care oferă lucrări de artă gratis, că e dificil să decideți de la care să porniți sau la care să vă opriți. Exemplele din această secțiune reprezintă doar o mică parte din multitudinea de situri disponibile. Numai indexul se întinde pe cinci pagini – o colecție gigantică de lucrări de artă pentru paginile Web. Nu depinde de anumite platforme.

Programe plug-in din diverse surse

Există practic zeci, poate chiar sute de programe plug-in speciale, disponibile pentru browserele mari, fiecare dintre ele având un anumit rol, dar fiecare dintre ele vă limitează auditoriul doar la aceia care au acel plug-in instalat sau nu-i deranjează să-l încarce pe hard-disc. Dacă lucrarea dumneavoastră necesită unul dintre programele plug-in, deoarece trebuie să distribuiți un anumit format de fișier sau să dați un anumit aspect, folosiți-le cu orice preț, dar cu cât vă apropiați de formatele „obișnuite" care sunt acceptate de browserele principale, cu atât veți avea o audiență generală. Programele plug-in necesită timp pentru descărcare și spațiu pe hard-disc și mulți utilizatori nu au nici timp, nici spațiu pe disc.

Puteți găsi o listă bună cu programe plug-in disponibile pentru Netscape, la următoarea adresă:

`http://home.netscape.com/comprod/products/navigator/version_2.0/plugins/index.html`

Multe dintre aceste programe au și versiunile de Internet Explorer.

Multe dintre siturile pentru descărcare menționate în secțiunea anterioară, despre unelte grafice, conțin liste lungi de programe plug-in. Încercați-le și pe ele.

Multimedia pentru profesioniști

Când vă gândiți la Multimedia pe Web, primele două lucruri care vă trec prin minte sunt Macromedia și RealAudio, ale căror produse Shockwave, Director și RealAudio aproape definesc pentru majoritatea utilizatorilor termenul *multimedia*. Deși toate aceste tehnologii funcționează bine la conexiuni de viteze mari, ele pot genera fișiere mici care se încarcă repede, permițând conținutului multimedia să fie afișat sau să fie auzit înainte ca programul de vizionare să renunțe. Ambii furnizori oferă programe de vizionare gratis pentru publicul larg (poate împreună cu o versiune „premium" plătită) și vând instrumentele necesare pentru a crea propriile formate de date. Un avantaj important al acestor formate este dat de faptul că aproape toți utilizatorii care sunt interesați de multimedia pe Web, le au deja descărcate și instalate. Unele browsere vin deja cu aceste două formate preinstalate. Tabelul G.7 prezintă unele dintre cele mai bune instrumente și produse pentru o utilizare profesională a sitului Web.

Tabelul G.7. Multimedia pentru profesioniști.

Produs	*Tip/Caracteristici*	*Audiență*	*Gama de preț*	*Platforme*
RealAudio și RealVideo	Server audio/video	Profesioniști	$$$$$	WinNT UNIX
Macromedia	Instrument de creație Animație	Profesioniști	$$$$	Win95/NT Mac
Asymetrix	Instrument de creație Animație Educație	Profesioniști	$$$	Win95/NT
3D Studio Max	Proiectare grafică Redare 3-D Animație în 3-D	Profesioniști	$$$$	WinNT

RealAudio/Video

`http://www.realaudio.com`

Prezentări pe ieșiri audio-video și un furnizor de conținut, oferind un „radio" Web, cât și SSTV (slow-scan TV). Are disponibile programe plug-in gratis și contra cost. Pachetul server este cam scump, dar vă poate introduce în transmisia pe Web, fiind un pachet „la cheie". Această companie creează de asemenea utilitare gratuite pentru conversia formatelor audio uzuale, în formatele particulare pentru transmisie prin canale de ieșire. Fișierele rezultate pot fi tratate de orice server HTTP ca fișiere de pseudo transmisie, care va fi recunoscut de cei care îl redau. Rulează pe platformele Macintosh și Win95.

Macromedia

`http://www.macromedia.com/`

Sunt faimoasele îmbunătățiri „Shocked" pentru multe situri Web, oferind posibilitatea conținutului interactiv, jocuri și altele. Ei oferă de asemenea versiuni de probă ale unor programe, cât și versiuni la preț scăzut, oferind o parte din aceleași funcțiuni, așa că nu

renunțați la ele doar fiindcă acestea „costă prea mult". Dacă le încercați puțin, vi se va deschide apetitul pentru mai mult. Zeci de produse, inclusiv produsele fanion Director 6 Multimedia Studio și Authorware 4 Interactive Studio. Există și produse mai accesibile, cum ar fi Flash 2 și Backstage Internet Studio-Desktop Edition. Cele mai noi îmbunătățiri sunt ShockWave pentru transmisii, transmisii Internet directe și Shocked CD. Rulează pe platformele Macintosh și Win95.

Asymetrix Multimedia Toolbook

`http://www.asymetrix.com/`

Este o companie multimedia specializată în învățarea online, având unelte și experți pentru crearea de cursuri interactive pentru profesori sau rețelele intranet ale companiilor. Dacă aveți nevoie de acest tip de cunoștințe, Asymetrix este un bun loc de început. Ei oferă unelte pentru a crea imagini tridimensionale pe Web, proiectare Java vizuală pe loc, producție video digitală, un limbaj de scriptare pentru cursuri și alte utilitare pentru a susține cursuri pe Web. Unele sunt disponibile ca programe demonstrative. Rulează pe platformele Win95/NT.

3D Studio Max de la Autodesk/Kinetix

`http://www.ktx.com/`

Fiți pregătit pentru un șoc mai sever de astă dată, cu un preț de patru cifre, dar incredibil de puternic. Este genul de software la care visați când doriți să creați animație sau video în 3D la nivel de studio de film. Redarea este superbă, rivalizând cu stațiile grafice UNIX specializate, iar setul de facilități este foarte bogat, permițând utilizatorului norocos să-și vadă proiectul din cele patru unghiuri, selectându-l pe cel mai potrivit pentru o anumită operație.

Sunt disponibile programe plug-in pentru a realiza operații cum ar fi mișcare bipedă sau mișcarea mușchilor, așa că este un mediu potrivit pentru efecte cu adevărat spectaculoase și elemente de animație. Chiar dacă nu vi-l puteți permite, este genul de resursă pe care o cereți unei firme de proiectare grafică atunci când doriți o „animație foto realistă" pe situl dumneavoastră. Orice utilizator care are acest instrument pe calculatorul său este serios în privința proiectării grafice și animației în 3D. Există un CD demo disponibil pe situl Web indicat. Este recomandabil să aveți un procesor perfor- mant Pentium Pro (preferabil cu mai mult de un singur procesor și multă memorie RAM). Rulează pe platformele WinNT.

Multimedia pentru utilizări ocazionale

Dar cum rămâne cu noi, cei care nu ne putem permite instrumente de o mie de dolari sau mai mult? Există câteva tehnologii ieftine pentru crearea cel puțin a sunetului, incluzând fișierele `.WAV`, `.AU` și `.MID`, care au „programe de vizionare" încorporate în browserele importante. Când conținutul este mic ca dimensiune, timpii de încărcare, nu mai par mai mari decât timpii tehnicilor de transmisie prin canale de ieșire. Încă o dată, utilitarele shareware au mult de oferit, dar dumneavoastră adesea sunteți tributari costului redus al unui mediu mai puțin integrat, care necesită mai mulți pași în diverse programe, pentru a realiza o anumită operație. Tabelul G.8 vă oferă o prezentare a resurselor bune de grafică testate și evaluate și multimedia shareware.

Tabelul G.8. Mulimedia pentru utilizări ocazionale.

Produs	*Tip/Caracteristici*	*Audiență*	*Gama de preț*	*Platforme*
MacWorld	Listă de resurse	Toți utilizatorii	Gratis	Mac
PCWorld	Listă de resurse	Toți utilizatorii	Gratis	Win95/NT
Hotfiles	Listă de resurse	Toți utilizatorii	Gratis	Mai multe
Sound Forge	Editor de sunet	Profesioniști	$$$	Win95/NT
NoteWorthy	Compozitor MIDI	Profesioniști	$$	Win95/NT
GoldWave	Editor de sunet	Toți utilizatorii	$$	Win95/NT
WebShot	Video în direct	Toți utilizatorii	$$$	Win95/NT
Movie Cleaner	Utilitar pentru filme	Toți utilizatorii	$$	Mac
RealAudio	Codificator audio	Toți utilizatorii	Gratis	Win95
RealVideo	Codificator video	Toți utilizatorii	Gratis	Mac
Personal AVI	Editor video	Toți utilizatorii	$$	Win95
Sirius Multimedia Links	Listă de resurse	Toți utilizatorii	Gratis	Mac
Cross Platform	Listă de resurse	Toți utilizatorii	Gratis	Windows
3DSite	Listă de resurse	Toți utilizatorii	Gratis	Mai multe
WDVL	Listă de resurse	Toți utilizatorii	Gratis	Mai multe
Graphics	Listă de resurse	Toți utilizatorii	Gratis	Mai multe
Grafica	Listă de resurse	Toți utilizatorii	Gratis	Mai multe
Obscura	Informații utile			

MacWorld

`http://www.macworld.com/`

Este o sursă bună de software pentru platformele Mac. Sunt disponibile unele recenzii software pe pagina index, dar majoritatea sunt oferite fără evaluări. Motoarele de căutare găsesc recenziile relativ ușor, iar o listă a celor mai bune programe de descărcat vă oferă o idee despre ce cred utilizatorii că merită. Rulează pe platformele Macintosh.

PCWorld

`http://www.pcworld.com/`

Este un sit asemănător cu MacWorld, oferind recenzii și evaluări despre hardware și software pentru platformele Windows/Intel. Același proprietar realizează aceste situri, nu chiar clone, dar foarte apropiate ca și concepție. Rulează pe platformele Windows/Intel.

Hotfiles Software Library de la ZDNet

`http://www.hotfiles.com/graphics.html`

Conține o selecție bogată de recenzii și evaluări despre shareware și demoware, bine organizate pe categorii. Rulează pe mai multe platforme.

Iată câteva exemple din siturile precedente, prezentând cam ce este disponibil, dar vă rog rețineți că sunt sute de programe în oricare din categoriile date. Ceea ce este prezentat în următoarele secțiuni reprezintă doar un mic eșantion din ceea ce e cel mai bun.

Sound Forge

`http://www.sfoundry.com/`

Este un editor de sunet de performanță ridicată pentru combinarea pistelor audio. Puțin cam scump, dar versiunea Sound Forge XP este mai accesibilă. Rulează pe platformele Windows/Win95.

NoteWorthy Composer

`http://www.ntworthy.com/`

Folosind acest program puteți crea propriile fișiere MIDI cu un compozitor MIDI ieftin. Rulează pe platformele Windows/Win95.

GoldWave

`http://www.goldwave.com/`

Înregistrator de sunet și editor shareware pe care îl puteți lua în considerare dacă Sound Forge este prea scump. Interfața cu utilizatorul nu este la fel de plăcută ca și cea de la Sound Forge, dar vă puteți permite să-l cumpărați mai mult decât Sound Forge. Rulează pe platformele Windows/Win95.

WebShot

`http://www.cinecom.com/products/webshot.htm`

Instalați-vă propria cameră video în direct pentru Internet! Uimiți-vă prietenii! Dacă ați dorit vreodată să expuneți detaliile biroului sau ale casei dumneavoastră pe World Wide Web, aceasta este șansa dumneavoastră. Rulează pe platforma Win95/NT.

Movie Cleaner Pro

`http://terran-int.com/`

Comprimă și aranjează fișierele video pentru prezentarea pe Web. Rulează pe platformele Macintosh.

RealAudio and RealVideo Encoders

`http://www.realaudio.com/products/encoder/`

Produceți propriile performanțe multimedia pe Web folosind formatele RealAudio și RealVideo, cât și un pseudo transfer continuu de pe orice server Web. Utilitare gratuite convertesc fișierele sursă uzuale în format „Real". Rulează pe platformele Win95 și Macintosh.

Personal AVI Editor

`http://www.flickerfree.com/`

Este un editor video pentru fișiere de tip AVI. Rulează pe platformele Windows.

Sirius Multimedia Links

`http://ardvark.com/mm.htm`

Conține legături de la utilizatori care sunt implicați în afaceri cu multimedia, așa că ei știu despre ce vorbesc; totuși, pagina este în mod evident demodată. Este un bun punct de pornire pentru orice căutare multimedia. Nu depinde de nici o platformă.

The Cross Platform Page

`http://www.mcad.edu/Guests/EricB/xplat.html`

Conține o colecție excelentă de legături multimedia cu un accent pe translatarea formatelor de fișiere între platforme. Schimburile dintre unele formate sunt clar explicate, așa că această pagină poate servi ca o scurtă introducere pentru unele programe de vizionare. Rulează pe toate pltaformele.

3DSite

`http://www.3dsite.com/`

O pagină grozavă cu resurse pentru proiectarea 3D pe Web și informații VRML. Resursele online (mai ales pentru Mac), listele de locuri de muncă și reclamele clasificate fac din sit un loc de întâlnire interesant pentru artiștii și animatorii Web. Mai ales pentru platforme Macintosh, dar are și resurse pentru alte platforme.

The World-Wide Web Virtual Library – Computer Graphics and Visualization

`http://www.dataspace.com/WWW/vlib/comp-graphics.html`

Conține destule legături către companiile comerciale, oferind instrumente grafice și multimedia pentru a le vizita, studii de cercetare online și alte materiale de vârf pentru grafică profesională. Nu depinde de anumite platforme.

Grafica Obscura de la Paul Haerberli

`http://www.sgi.com/grafica/`

Este un sit interesant al unui artist pe Web care prezintă exemple din lucrările lui, instrumente spectaculoase, articole, manuale pentru crearea de grafică și multimedia pentru Web, cât și genul de marfă pe care ați putea să o găsiți prăfuită într-un colț al unui studio. Foarte simple, dar foarte drăguțe. Unele dintre ele sunt incredibil de spectaculoase, în ciuda prezentării lineare fără sens. Nu depinde de o anumită platformă, deși este asociat cu SGI.

Limbaje de scriptare/independente de platformă

Scriptarea reprezintă partea dificilă, a proiectării unei pagini Web, unde mulți creatori cer ajutorul unui programator. Este dificil să stăpâniți perfect fiecare tehnică necesară pentru crearea unei pagini Web moderne, tot așa cum doar câțiva oameni pot construi o mașină începând de la șuruburi în ziua de azi. De fapt, mulți oameni talentați, proiectanți competenți, refuză să se uite „sub capota" paginilor Web și realizează numai partea de suprafață, lăsând partea de meșterit în seama „mecanicilor Web".

Nu este din cauză că scriptarea este imposibil de învățat; de fapt, următoarele referințe, din Tabelul G.9 vă vor arăta că multe operații sunt surprinzător de ușoare. Totuși, deși există timp pentru atât de multe lucruri de învățat, unii oameni trag linie atunci când vine vorba de orice fel de programare.

Tabelul G.9. Limbaje de scriptare/independente de platformă.

Limbaj	*Tip/Caracteristici*	*Audiență*	*Gama de preț*	*Plaforme*
Java	Compilat	Programatori	Gratis	Oricare
JavaScript	Interpretat	Toți utilizatorii	Gratis	Oricare
Jscript	Interpretat	Utilizatorii de IE	Gratis	IE
VBScript	Interpretat	Utilizatorii de IE	Gratis	IE

Java

Limbajul Java este răspunsul firmei Sun Microsystem la dominația crescândă a calculatoarelor Intel/Windows. Este un limbaj complet independent de platformă, care va rula binișor pe orice PC care conține un program „Java Virtual Machine", un anumit tip de interpretor ce se pretinde a fi un mecanism ideal pe care îl poate avea toată lumea, spre deosebire de multele combinații diverse între hardware și sisteme de operare, care există la ora actuală. Puteți crea, folosi și distribui miniaplicații Java aproape gratis, într-o configurație necomercială, deși aveți nevoie de o licență necomercială, pe care firma Sun v-o poate oferi gratis. Dacă totuși, intenționați să incorporați interpretorul într-un produs comercial, trebuie să cumpărați o licență comercială, caz în care trebuie să vă pregătiți să cheltuiți ceva bani.

Limbajul Java este un limbaj foarte productiv pentru designeri, deoarece tipizarea puternică și structurile de control excelente îl fac capabil să creeze un cod curat care să ruleze imediat corect. În completare, mecanismul „garbage collection" pentru memoria nefolosită îndepărtează una dintre principalele surse de erori pentru programatorii în C/C++ - pierderile de memorie și referirile incorecte la locațiile de memorie returnate sistemului din greșeală.

Unul dintre cele mai spectaculoase proiecte realizate în limbajul Java îl reprezintă programele servlets, programe Java pe partea serverului, care pot înlocui parțial scripturile CGI pentru siturile foarte accesate. Deoarece un servlet este întotdeauna rezident în memorie, supraîncărcarea implicată în începerea unui nou proces pentru fiecare cerere este eliminată și, deoarece Java Virtual Machine execută instrucțiunile Java mult mai repede decât este interpretat limbajul Perl pe majoritatea serverelor, creșterea medie a performanței poate fi extraordinară. Chiar mai important, deoarece un servlet rulează pe server, nu apar probleme la folosirea limbajului Java cu orice browser, indiferent dacă un mecanism Java este disponibil pe partea clientului.

Tabelul G.10 prezintă un scurt sumar despre tipul de informații pe care le puteți găsi pe Web, pentru a vă ajuta să scrieți cod în Java sau să găsiți cod Java deja scris pentru a rezolva o anumită problemă. Deși resursele sunt furnizate gratis, ele pot indica spre un sit comercial sau shareware cu etichete de preț atașate de codul lor, așa că fiți prudenți cu ce folosiți și cum folosiți. Nu este un procedeu neobișnuit ca prețurile licențelor software să varieze în funcție de scopul pentru care este folosit codul.

Tabelul G.10. Resurse Java

Sit	*Tip/Caracteristici*	*Audiență*	*Gama de preț*	*Plaforme*
Sun Java Home	Tutorial Exemple Furnizori	Programatori	Gratis	Oricare
Gamelan	Tutorial Exemple Furnizori	Programatori	Gratis	Oricare
Haifa Research	Mecanism Servlet	Programatori	Gratis	Oricare
Sun Servlets	Tutorial	Programatori	Gratis	Oricare
Programmers Source	Tutorial Exemple	Programatori	Gratis	Oricare

Java Home Page de la Sun

`http://java.sun.com/`

Este sursa pentru cele mai noi știri despre Java și tehnologiile aferente (aprobate de companie), cât și software pentru descărcare (Software Development Kit și exemple). Din construcție, nu depinde de anumite platforme.

Gamelan

`http://www.gamelan.com/`

Este un sit Web devotat limbajelor Java și JavaScript, deși este greu de imaginat exact, cum o orchestră gamelan (percuție și gong) este legată de Java, altfel decât prin faptul că provine din aceeași parte a lumii unde se face cafeaua pentru cunoscători. Presupun că este una dintre acele aluzii obscure de care sunt atașate persoanele cu gusturi sofisticate. Cu toate acestea, este o resursă grozavă. Nu depinde de platformă.

Haifa Research Institute de la IBM

`http://www.haifa.il.ibm.com/servlet_express/`

Este un sit al companiei IBM, care conține un servlet Java ce poate fi descărcat, Servlet Express. Programele servlet sunt foarte actuale în lumea Java deoarece ele permit unui proiectant să adauge funcționalități de tip CGI direct pe server, fără să mai înceapă un nou proces pentru fiecare cerere. Servlet Express este un mecanism care permite unui proiectant să extindă funcționalitatea aproape a oricărui server dar în special a acelor servere care nu au suport încorporat sau a celor care au suportul pentru servlet învechit. Este o tehnologie grozavă; încercați-o. Nu depinde de anumite platforme.

Servlet Page de la Sunsoft

`http://jserv.javasoft.com/products/java-server/servlets/index.html`

Sursa definitivă de informații referitoare la programe servlet Java, Servlet Development Kit (SDK), ce poate fi descărcată, permite designerilor să conceapă și să ruleze rapid folosind această tehnologie nouă. Legăturile către medii comerciale de dezvoltare care acceptă programe servlet permit chiar și neprogramatorilor să achiziționeze componente gata realizate pentru a permite oricărui proiectant Web să profite de viteza și independența de platformă a programelor servlet pentru realizarea proiectelor Web. Nu depinde de anumite platforme.

The Programmer's Source–Java Resources

`http://www.progsource/java/`

Este o listă extinsă de legături la manuale, pe o singură pagină, la resurse de dezvoltare și alte legături interesante pentru profesioniștii în Java. Nu depinde de anumite platforme.

JavaScript

Limbajul Java are o mare problemă. Trebuie să știți să programați pentru a-l folosi și aveți nevoie de un compilator și alte programe care nu se prea găsesc pe majoritatea calculatoarelor, deși firma Sun vă va da unele programe gratis. Acesta este un obstacol de trecut pentru majoritatea utilizatorilor (deși Java este mai ușor de învățat decât C sau C++), așa că Netscape a inventat un limbaj asemănător cu limbajul Java, care nu are nevoie de nici un instrument special – doar de editorul cu care scrieți restul paginii – și care are un set de comenzi mult mai simple, așa că oricine poate să facă o „programare" utilă fără cunoștințe sau pregătire speciale. JavaScript are o slăbiciune; definiția pe care Netscape a dat-o limbajului nu include un mecanism de versiune pentru a contoriza modificările și diferențele

dintre implementări, ceea ce a cauzat probleme deoarece propria sa definire a evoluat și cere utilizatorilor să-și actualizeze browserele periodic, riscând altfel să rămână în urmă. O trecere în revistă a resurselor JavaScript o găsiți în Tabelul G.11.

Tabelul G.11. Resurse JavaScript.

Sit	*Tip/Caracteristici*	*Audiență*	*Gama de preț*	*Platforme*
Web Reference	Tutorial Exemple Furnizori	Toți utilizatorii	Gratis	Oricare
JavaScript	Tutorial Exemple Furnizori	Toți utilizatorii	Gratis	Oricare
JavaScript Planet	Tutorial Exemple Furnizori	Toți utilizatorii	Gratis	Oricare
JavaScript World	Tutorial Exemple	Toți utilizatorii	Gratis	Oricare
Cut-N-Paste JavaScript	Exemple	Toți utilizatorii	Gratis	Oricare
JavaScript Debugger	Instrument	Toți utilizatorii	Gratis	Oricare
Visual JavaScript	Instrument	Toți utilizatorii	Gratis	Oricare
JavaScript Index	Exemple	Toți utilizatorii	Gratis	Oricare
Snark Hunt JavaScript	Exemple	Toți utilizatorii	Gratis	Oricare
Voodoo JavaScript	Tutorial	Toți utilizatorii	Gratis	Oricare

JavaScript Page de la Web Reference

`http://www.webreference.com/javascript/`

Nu a mai fost actualizat de mult, este încă interesant, având manuale și alte informații. Nu depinde de o anumită platformă.

JavaScript

`http://javascript.developer.com/`

Este asemănător ca și concepție cu Gamelan, găzduit de asemenea pe situl `developer.com`, care conține subsituri dedicate altor limbaje și tehnologii, inclusiv ActiveX. Puteți să vă

legați la el direct de la Gamelan (la adresa `java.developer.com`), dar adresa URL dată aici reprezintă legătura directă. Nu depinde de o anumită platformă.

JavaScript Planet

`http://www.geocities.com/SiliconValley/7116/`

Este un sit mic, cu aspirații mari, un buletin informativ și sute de exemple de programe scrise în JavaScript. O caracteristică plăcută este avertizarea vizibilă de lângă scripturile care nu vor funcționa pe Internet Explorer, cât și alte incompatibilități de browsere/platforme. Nu depinde de o anumită platformă hardware.

JavaScript World

`http://www.jsworld.com.`

Acest sit popular conține o arhivă vastă, cât și știri și grupuri de dialog. Nu depinde de anumite platforme.

Cut-N-Paste JavaScript

`http://www.infohiway.com/javascript.`

Acest sit este specializat în programe JavaScript deja scrise, pe care le puteți copia în propriile pagini pentru a realiza anumite operații. Gama de scripturi, se întinde de la scripturi simple de text cu derulare până la calculatoare complexe și prezentări grafice cu diapozitive. Nu depinde de anumite platforme.

Netscape JavaScript Debugger

`http://search.netscape.com/eng/Tools/JSDebugger/relnotes/relnotespr1.html`

Este un instrument de uz general pentru depanare, ce poate ajuta utilizatorii să-și detecteze greșelile permițându-le să treacă pas cu pas prin cod și să examineze valorile. Nu depinde de anumite platforme.

Visual JavaScript 1.0 de la Netscape

Este un mediu de dezvoltare vizuală pentru JavaScript. Acest instrument puternic vă va ajuta să creați scripturi cu ajutorul unui editor de machetare cu capacități de tip drag-and-drop. Nu depinde de anumite platforme.

JavaScript Index

`http:/www.cob.ohio-state.edu/~lindeman/javascript/andrew.html`

Este o sursă bună pentru o mulțime de scripturi, de la calculatoare și programe de calcul tabelar, până la Web Prophet. Nu depinde de anumite platforme.

Hunting the SNARK with JavaScript

`http://www.cs.cmu.edu/~jab/snark/`

Acest sit vă arată cum să puneți un motor de căutare JavaScript pe un sit Web. Nu depinde de anumite platforme.

Introduction to JavaScript de la Voodoo

`http://rummelplatz.uni-mannheim.de/~skoch/js/tutorial.htm`

Manualul excelent al lui Stefan Koch prezintă cam tot ce puteți face cu JavaScript. Nu depinde de anumite platforme.

JScript

JScript de la Microsoft a fost o sursă de confuzii între cei care scriu cod în JavaScript de când Microsoft a început să meargă pe propriul său drum cu limbajul. Cea mai nouă versiune este mult mai corespunzătoare, dar încă mai are diferențe semnificative de care trebuie ținut cont la proiectarea pentru Internet. Între timp, deoarece mai sunt folosite multe versiuni mai vechi de Internet Explorer și vor mai fi folosite încă în continuare, codul trebuie fie să fie limitat la o submulțime compatibilă, fie Explorer ar tebui să fie testat explicit pentru – și codul să fie scris pentru a ține cont de - diferențe fără să arunce pe ecran mesaje de eroare.

Rulează pe platformele Win95/NT și eventual are suport pentru PowerMac. Terții pot furniza suport UNIX pentru JScript, deși este mai puțin important atât timp cât JScript devine tot mai mult JavaScript.

Jscript 2.0 Web Page de la Microsoft

`http://www.microsoft.com/jscript/`

Este o referință bună pentru kiturile de instrumente JScript și Microsoft; de asemenea include manuale și sfaturi pentru crearea scripturilor JScript pentru browserele IE. În general, nu se poate avea încredere în Microsoft JScript că nu va arunca pe ecran erori JavaScript când este vizionat de alte browsere, nici în JavaScript nu se poate avea încredere pe deplin că va rula pe IE fără a căuta prezența lui Microsoft Internet Explorer și fără a coda anumite caracteristici. Este păcat, deși aderența Microsoft la standarde se îmbunătățește tot mai mult în ultima vreme. Pe de altă parte, într-un mediu Microsoft exclusiv, cum ar fi o rețea intranet controlată de platformă, accesul la instrumentele proprii Microsoft este superb, deși VBScript este dovedit mult mai bun în acel mediu.

VBScript

Este răspunsul firmei Microsoft la JavaScript. Deoarece JavaScript și Jscript sunt parțial incompatibile, calea cea mai sigură pentru a reliza operații similare (cel puțin pentru Internet Explorer) este de a folosi VBScript, care este complet sub controlul Microsoft. Interfațează bine cu ActiveX și alte tehnologii Microsoft și este o alegere bună pentru rețelele intranet cu o configurație standard Win95/NT. Mulți proiectanți folosesc în pagină atât JavaScript, cât și VBScript, așa că pot forța ca IE să aleagă VBScript, pe care Netscape nu îl recunoaște, și să ignore imediat următorul JavaScript, dar aceasta înseamnă că unele funcții trebuie să fie codate de două ori.

Rulează pe platformele Win95/NT în principal, dar suportul pentru PowerMac de obicei urmează după ce versiunile Win95/NT au apărut pe piață de ceva timp. Terții s-ar putea să asigure suport pentru plaformele UNIX, dar aceasta nu se poate garanta.

VBScript 2.0 Web Page de la Microsoft

`http://www. microsoft.com/vbscript/`

Este o bună referință pentru VBScript, cu posibilități de descărcare gratis, manuale, documentație și alte surprize pentru a putea scrie mai ușor codul VBScript. Interfațează bine cu ActiveX și alte tehnologii Microsoft. Rulează pe platformele Win95/NT.

Validatori HTML

Acum că ați proiectat și scris codul pentru pagina dumneavoastră, aceasta funcționează? Este corectă? Cum puteți avea încredere în verificarea de validitate, dacă aceasta există, realizată de programul de editare? Vă va spune dacă ceva nu a mers bine? Nu strică

niciodată să vi se spună o a doua părere sau chiar o a treia. Eu, de obicei, testez folosind cel puțin doi sau trei validatori și, de asemenea, deschid paginile mele pe mai multe browsere diferite.

Doar fiindcă un validator vă spune că o combinație de etichete nu trece de verificarea unui anumit standard, nu înseamnă că pagina dumneavoastră nu este bună. Uneori trebuie să folosiți „extra" etichete sau atribute pentru a vă asigura că paginile dumneavoastră arată aproape la fel pe mai multe browsere diferite, dar validatorul îl indică ca și cum ar fi „necorespunzător". Nici unul dintre validatori nu cunoaște nimic despre „talentele" implicate în echilibrarea cerințelor multelor „standarde" în conflict de pe Web, așa că sfatul meu este de a nu lua mesajele de avertizare ca atare, atât timp cât treceți testele de validitate de bază. Tabelul G.12 rezumă câteva dintre instrumentele de validare și resurse.

Tabelul G.12. Validatori HTML.

Produs	*Tip/Caracteristici*	*Audiență*	*Gama de preț*	*Platforme*
CSE 3310	Desktop Multe opțiuni	Programatori	$$	Win95/NT
WebTechs	Online/ Desktop Multe opțiuni	Programatori	Gratis	N/A sau UNIX
Weblint	Online/ Desktop Multe opțiuni	Programatori	Gratis	N/A sau mai multe
Bobby	Online/ Desktop Multe opțiuni	Programatori	Gratis	Mai multe sau oricare

CSE 3310 HTML Validator 2.0

`http://www.htmlvalidator.com/`

Acceptă în mod curent definiții DTD până la HTML 3.2 plus extensiile neprecizate de la Netscape și Microsoft. Este un validator independent flexibil, pentru folosirea pe desktop. Rulează pe platformele Win95/NT.

WebTechs Validation Service

`http://www.webtechs.com/html-val-svc/`

Este o resursă online pentru validarea fie a unei pagini întregi, fie a unei bucăți de cod folosind un formular. Include mai multe opțiuni, cum ar fi interpretarea exactă a standardelor IETF 2.0, IETF 3.0 (acum retras), W3C 3.2 (Wilbur), W3C 3.2 + CSS1 și W3C 4.0 (în proiect). Acceptă de asemenea definiții DTD care descriu, mai mult sau mai puțin, comportarea extensiilor particulare pentru Mozilla (Netscape Navigator), SoftQuad, AdvaSoft, Microsoft IE și Microsoft IE 3.0 Beta. Foarte „temperamental" când întâlnește anumite tipuri de erori. Există pe Web diverse intrepretări ale aceluiași cod, inițial un serviciu de la Hal Software (Dave?), iar multe sunt disponibile gratis pentru a fi descărcate. WebTechs furnizează kitul HTML Check Toolkit, care este o versiune gratuită cu linie de comandă a programului pentru sistemele UNIX. Nu depinde de anumite platforme.

Weblint de la Neil Bower

`http://www.cre.canon.co.uk/~neilb/weblint/`

Este un script Perl care „adună scamele de pe paginile Web" și este gratuit pe Internet. Aceasta este una dintre cele câteva versiuni, dar toate au o moștenire comună de la bunica tuturor „adunătorilor de scame", Lint, care a fost creat pentru cod C pe sistemele UNIX.

Weblint nu „știe" limbajul HTML și nu oferă rapoarte extinse ale erorilor pentru serviciul WebTechs, dar este mai robust prin faptul că păstrează mesajele importante generate chiar și după ce se împiedică de o greșeală serioasă. Mulți validatori încep să genereze rapid pagină după pagină cu mesaje de eroare relativ nesemnificative, determinate de lipsa unei etichete sau a unei ghilimele. Acest sit conține o listă foarte bună a altor resurse de validare și codare, inclusiv porți Web spre Weblint însuși. Rulează pe orice platformă care acceptă un interpretor Perl, inclusiv UNIX, Win95/NT și Macintosh.

Bobby 2.0 de la CAST

`http://www.cast.org/bobby/`

Acesta este un validator special care testează pagina pentru conformitatea cu regulile de accesibilitate, cât și aderența la standardele externe. Este mai greu decât v-ați putea închipui dar, după ce ați trecut de testul Bobby, puteți plasa micul logo drăgălaș cu Bobby pe pagina dumneavoastră, ca o certificare a testării paginii. Spre deosebire de majoritatea altor validatori, Bobby validează conformitatea cu WebTV, Lynx și AOL și extinde caseta de validare cu „proprietary extensions" care se găsește în majoritatea programelor de validare externe pentru a preciza versiunile de IE și Navigator. Din nefericire, încă nu acceptă IE 4.0 sau Navigator 4.0.

Ca proiectanți, avem responsabilitatea socială să ținem cont de nevoile persoanelor cu handicap, dar accesibilitatea pe Web va deveni o garantare legală pentru afacerile mari și alte structuri guvernamentale din SUA și din alte țări, așa cum acesibilitatea în locurile publice este garantată prin acces fizic și la telefon. Fiind atent la aceste probleme acum, ați putea salva clientul sau afacerea de un proces mai târziu și vă dă sentimentul că ați făcut un lucru bun și corect.

Sursa de la Bobby 1.2.1 este disponibilă pentru descărcare sub forma unui script Perl, așa că ar trebui să poată fi rulat pe orice sistem care are un mecanism Perl, deși ei promit doar compatibilitate UNIX. S-ar putea să trebuiască să modificați scriptul pentru a elimina dependențele UNIX. O variantă a limbajului Java din versiunea 2.0 care va rula fără să fie nevoie să modificați scriptul, ar trebui să fie disponibilă în momentul în care citiți această carte. Este un produs cu adevărat spectaculos și este gratis, așa că îl puteți considera o donație. Nu depinde de o anumită platformă.

Servere Web

Un server Web este ceea ce stă pe serverul dumneavoastră sau pe serverul furnizorului dumneavoastră de servicii și care răspunde la cererile făcute de browserul dumneavoastră sau de multe browsere ale persoanelor care vă vizitează situl. Ei formează o echipă cu dumneavoastră sau cu altă persoană (de obicei) pe de o parte și de cealaltă parte datele care doriți să ajungă la ceilalți, iar acești doi agenți cooperează încercând să realizeze transferul dintr-o parte în alta și invers.

Cooperarea este posibilă datorită unui limbaj special pe care ambii îl cunosc, HyperText Transfer Protocol (HTTP), care este schema ce face ca datele și conținutul să fie disponibile pe suprafața de lucru a calculatorului dumneavoastră, ca și cum ar fi chiar în interiorul lui, când de fapt ele ar putea să fie în partea celalată a lumii. Tabelul G.13 rezumă câteva dintre cele mai populare servere Web.

Tabelul G.13. Servere HTTP.

Produs	*Tip/Caracteristici*	*Audiență*	*Gama de preț*	*Plaforme*
CERN HTTPD	HTTP 1.0	Expert	Gratis	Mai multe
NCSA HTTPD	HTTP 1.0	Expert	Gratis	Mai multe
NCSA HTTPD/Win	HTTP 1.0	Expert	Gratis	Windows
ORA WebSite	HTTP 1.1	Intermediar	$$$	Win95/NT
MacHTTP	HTTP 1.0	Intermediar	$$	Mac
WebStar	HTTP 1.1	Intermediar	$$$	Mac
Microsoft	HTTP 1.1	Intermediar	$$$$	Win95/NT
Netscape	HTTP 1.1	Intermediar	$$$$	Win95/NT UNIX
Jigsaw	HTTP 1.1	Expert	Gratis	Oricare
CL-HTTP	HTTP 1.1	Expert	Gratis	Linux, Mac
Apache	HTTP 1.1	Expert	Gratis	Unix, Win95/NT
Sun Servers	HTTP 1.1	Expert	$$$$	Sun UNIX Solaris
Lotus Domino	N/A	Expert	$$$$$	N/A
Cold Fusion AS	N/A	Intermediar	$$$	WinNT

CERN HTTPD

`http://www.w3.org/pub/WWW/Daemon/`

Acesta este faimosul server CERN care a constituit, împreună cu serverul NCSA, baza pentru o mare parte din rețeaua Web inițială și conduce încă o mare parte din ea. Ultima lansare a fost în anul 1996, iar toată munca din viitor va fi pentru serverul Jigsaw Java, care va fi tratat mai târziu în acest capitol. Rulează pe platformele UNIX, dar a fost introdus cu numele schimbat pe alte câteva platforme.

NCSA HTTPD

`http://hoohoo.ncsa.uiuc.edu/index.html`

Acesta este celălalt nume „de marcă" în lumea serverelor. Amândouă aceste servere Web obișnuite oferă o contribuție de bază de servicii și sunt bine cunoscute. La fel ca și cum ați cumpăra un Ford sau un Chevy, piesele de rezervă sunt ușor de procurat și există o mulțime de „mecanici" Web care știu să le înlocuiască. Sunt o „familie" stabilă, plăcută de servere. Rulează de asemenea, pe platformele UNIX, dar există portabilitate/recodificare pentru Windows (vedeți secțiunea următoare).

NCSA HTTPD for Windows

`http://tech.west.ora.com/win-httpd/`

Aceasta este probabil cea mai nouă actualizare sub Windows pentru acest software. Nu mai este complet freeware, dar este un produs mai mult sau mai puțin comercial oferit de O'Reilly, care a fost înlocuit de WebSite (vedeți secțiunea următoare). Rulează doar pe platformele Windows.

WebSite 2.0 de la O'Reilly

`http://website.ora.com/`

Este un server Web pe 32 de biți care conține toate accesoriile, tranzacții comerciale sigure și alte facilități plăcute care vă ajută să lansați un server Web adevărat într-un timp record, fără a fi nevoie de o mulțime de încercări. Acesta este serverul O'Reilly la care se lucrează acum și la care vă puteți aștepta să vedeți unele îmbunătățiri în timp. O dezvoltare aparte interesantă o reprezintă suportul incorporat pentru programele servlet Java din versiunea WebSite Pro 2.0. Rulează pe platformele Win95/NT.

MacHTTP

`http://www.starnine.com/machttp/`

Este un server Web shareware pentru Mac OS, înlocuit de WebStar de la aceeași firmă, care este mult mai rapid și supus unei dezvoltări active. Pagina MacHTTP încă se laudă cu compatibilitatea cu Mosaic, deci evident nu prea depune efort pentru a-l actualiza. Rulează pe platformele Macintosh.

WebStar 2.1

`http://www.starnine.com/`

Este un server modern cu îmbunătățiri, cum ar fi utilizarea rapidă și dinamică a memoriei cache. Integrat de Apple în pachetul Apple Internet Server Solution Version 2.1, WebStar este un server HTTP complet echipat pentru lumea Apple. Rulează pe platformele Macintosh.

Microsoft Web Servers

`http://www.microsoft.com/products/os.htm`

Conține diverse servere Web ale companiei Microsoft. Toate sunt foarte bine integrate cu produsele Microsoft, dar pot fi dificil de administrat pentru situri foarte mari și s-a spus că au o interfață Microsoft incomodă cu celelalte resurse Web. Totuși, când rulați un sit Microsoft/Intel, acesta este probabil cel mai ușor de folosit, deoarece se potrivește bine cu orice altceva, iar dumneavoastră nu mai sunteți nevoit să învățați un nou sistem de operare. Rulează pe platformele Microsoft WinNT, fie Intel sau DEC Alpha.

Netscape SuiteSpot Servers

`http://www.netscape.com/comprod/server_central/`

Este sursa centrală pentru întreaga rețea de servere Netscape și instrumente de dezvoltare axate pe Web, inclusiv Visual JavaScript, NetObjects Fusion, Symantec Visual Café Pro și alte instrumente pentru serverele și clienții Netscape. Ei au mulți parteneri care oferă produse ce extind funcționalitatea serverelor lor, oferind facilități cum ar fi scanarea virușilor la serverul proxy sau la modulul firewall, motoare de căutare, instrumente pentru baze de date, cât și utilitare administrative. Rulează pe platformele UNIX și WinNT.

Jigsaw

`http://www.w3.org/Jigsaw/`

Este un server complet echipat, produs de W3C, care înlocuiește serverul CERN HTTPD în preferințele celor de la W3C și în efortul lor de dezvoltare. Este scris în totalitate în limbaju Java, deci va rula pe orice calculator care are un mecanism virtual Java. Jigsaw acceptă de asemenea programe servlet Java, echivalentul pentru server a miniaplicațiilor Java, cât și o modalitate mai rapidă de executare a operațiilor de tip CGI cu mai puțină bătaie de cap.

Versiunea curentă este doar în cod alfa, dar măcar este ceva de la care se poate porni. Rulează, din proiectare, pe orice platformă.

Common Lisp Hypermedia Server (CL-HTTP)

`http://www.ai.mit.edu/projects/iiip/doc/cl-http/home-page.html`

Este un server Web scris în limbajul Common Lisp, având codul sursă disponibil gratis pentru platformele Linux și Macintosh. Este o resursă excelentă pentru programatori și alți utilizatori care vor să experimenteze cu adevărat serverele și să împingă limitele la maxim. Rulează pe platformele Linux și Macintosh.

Apache

`http://www.apache.org/`

Este un server HTTP freeware care conține opțiunea SSL. La fel ca majoritatea programelor freeware, din mediul de proiectare deschis, lucrurile sunt în continuă schimbare în funcție de timpul liber pe care îl are la un moment dat comunitatea disparată de autori. Rulează pe platformele UNIX, dar portabilitatea pentru WinNT doar pentru codul sursă va fi lansată în viitor.

Sun Server Family

`http://www.sun.com/servers/`

Este o linie de servere HTTP extrem de capabile, produsă de firma Sun, destinată să solicite cât mai puțin posibil intervenția utilizatorului și care rulează pe majoritatea casetelor lor de la linia NetraJ la Ultra Enterprise Server. O linie SPARC tolerantă la greșeli este îndreptată în special spre industria telecomunicațiilor sau către oricine care are nevoie de o adevărată fiabilitate. Integrarea superbă cu limbajul Java face din această linie o alegere bună pentru magazinele Internet bazate pe Java, dar serverele vor rula doar pe platformele Sun UNIX, așa că aceia care au fobie la UNIX nu vor fi satisfăcuți. Sunt totuși produse grozave și vor mulțumi desigur pe cei mai tehnici dintre dumneavoastră. Rulează doar pe platformele Sun UNIX.

Lotus Notes/Domino

`http://www.lotus.org/`

Ar trebui să spun câteva cuvinte despre IBM/Lotus și Domino aici, cu toate că ele nu sunt de competența majorității utilizatorilor acestei cărți, deoarece interfața lor browser este cel puțin parțial (în majoritate?) particularizată, iar platforma necesită un suport de corporație și angajamente de mare desfășurare. Viziunea lor de ansamblu se apropie de cea a unei combinații server/browser Web, cu toate că, și ar fi o neglijență din partea mea să nu menționez cel puțin în treacăt, și tendința firmei IBM către limbajul Java i-ar putea face să despartă conceptul de corporație în sisteme mici de afaceri. Modelul de realizare a afacerii creat de la Lotus Notes/Domino este atoatecuprinzător, incluzând „capriciile" altor browsere și servere particulare, cu un mediu bine definit, în care extensiile particulare pentru capacitățile standard sunt regula și nu excepția.

Dacă auditoriul dumneavoastră este localizat pe rețeaua intranet a unei mari corporații, organizații guvernamentale sau consorțiu, Lotus este adecvat deoarece integrează cu succes mesageria, conferințele, agendele, accesul la baze de date, cât și o mulțime de alte funcții pe care tot mai mulți cetățeni ai Internetului pot să le admire doar de la distanță. Nu numai atât, dar uneltele administrative au fost proiectate pentru situri care au mai mult de o mie de utilizatori, și nu sunt cârpite precum unele pe care nu vreau să le numesc. Nu spun că nu ați fi putut face toată munca mai ieftin și din mers realizând propria integrare și dezvoltare, dar cine are timp pentru așa ceva? Dependență completă de platforme.

Cold Fusion Application Server

`http://www.allaire.com/`

Este celălalt partener al parteneriatului Cold Fusion și nu este un server adevărat, chiar dacă l-am introdus aici. În ciuda numelui, CF Application Server nu este un server HTTP deloc, este mai mult un fel de interpretor (cum ar fi un CGI, ce va fi prezentat în secțiunea următoare) pentru etichete și atribute speciale create de Cold Fusion. L-am inclus la secțiunea de servere deoarece este cel puțin asociat mai mult cu serverele, putând să „pretindeți" că serverul HTTP înțelege puțin mai mult decât poate, preluând funcționalitatea mai multor scripturi și făcând conținutul lor disponibil pentru proiectantul Web într-un format de etichete și atribute speciale care ascund complexitatea ce stă la baza lor sub interfața familiară etichetă/atribut. De asemenea, precompilează paginile Cold Fusion pentru ca ele să fie afișate rapid. Consultați discuția precedentă despre Cold Fusion la secțiunea „Instrumente pentru situri Web: Publicarea pe Web și administratori de sit". Rulează pe platformele WinNT.

Common Gateway Interface (CGI)

Ei bine, acum că avem pagini Web răspândite în toată lumea, cineva s-a gândit că ar fi bine dacă ați putea vorbi cu serverul și să-i cereți ceva – lucruri care nu pot fi descrise ca fiind pagini și lucruri pe care doreați să le scrieți sau să le spuneți și nu să citiți sau să ascultați. Problemă simplă; HTTP a fost inițial o stradă cu sens unic. Puteați cere lucruri și le primeați fără probleme, dar HTTP nu putea urmări unde ați fost în diverse momente și nu putea stoca ceva ce a învățat.

Astfel a fost creat un nou protocol, Common Gateway Interface sau CGI care vă permite să scrieți cod special penru a cere și a primi aproape tot ce doriți de la server și creează încă un limbaj și set de instrumente pe care să le învățați. Tabelul G.14 conține referințe la instrumente generale pentru a alcătui cea mai mare parte a protocolului CGI.

Tabelul G.14. Resurse CGI.

Resurse	*Tip/Caracteristici*	*Audiență*	*Gama de preț*	*Plaforme*
NCSA CGI	Listă de resurse Tutorial	Programatori	Gratis	Oricare
CGI Resources	Listă de resurse Tutorial Instrumente	Programatori	Gratis	Oricare
CGI Collection	Listă de resurse Instrumente	Programatori	Gratis	Oricare
CGI Made Easy	Listă de resurse Tutorial Instrumente	Programatori	Gratis	Oricare
Matt's Script Archive	Listă de resurse Instrumente	Programatori	Gratis	Oricare

(continuare)

Tabelul G.14. – continuare

Resurse	*Tip/Caracteristici*	*Audiență*	*Gama de preț*	*Platforme*
Selena Sol's Script Archive	Listă de resurse Tutorial Instrumente Comentarii sociale	Programatori	Gratis	Oricare

Informații generale

Iată câteva informații utile despre CGI, informații care acoperă o gamă largă de elemente și instrumente, de la „Cum să programăm interfața CGI" până la „Iată ceva ce puteți folosi azi".

The Common Gateway Interface at NCSA

`http://hoohoo.ncsa.uiuc.edu/cgi/`

Este o introducere excelentă în CGI, de la cei care l-au inventat sau care au fost prin preajmă cel puțin, când a fost inventat, incluzând o introducere, un curs introductiv, manuale, specificații și alte informații pentru a putea începe rapid să programați interfața CGI.

CGI Resources

`http://www.halcyon.com/sanford/cgi/index.html`

Este o colecție bogată de manuale, unelte de depanare și alte resurse pentru începătorul în scrierea CGI. Se presupune că aveți cunoștințe de limbaj Perl, cel mai folosit limbaj de programare a interfeței CGI. Nu depinde de o anumită platformă, deși majoritatea porților CGI se află pe calculatoare sub UNIX, așa că ați putea afla unele informații în exemplele de cod reale.

The CGI Collection

`http://www.selah.net/cgi.html`

Este o colecție de resurse, micuță, dar bine organizată de la un furnizor de servicii din Statul Washington, `Selah.net`, care promovează World Wide Mart's CGI Resource Index pe pagina de deschidere. Adresele scripturilor accesează contoare, camere de discuții, parole de protecție și altele, dar manualele par să fie în afară. Cu atâta reclamă reciprocă, este greu de spus cui aparține în final o anumită pagină.

Majoritatea scripturilor actuale par să fie scrise în limbajul Perl, dar mai sunt unele scrise și în C. Una dintre caracteristicile plăcute este CGI-Bot, care va căuta informații în mai multe pagini cu resurse CGI. Ca multe dintre astfel de mici situri, conținutul pare să semene mai mult a pretext pentru a momi utilizatorii să privească reclamele și nu cu rezultatele unei munci pasionate cum erau siturile CGI mai vechi. De necontestat, există și lucruri valoroase în pagină. Rulează pe platformele UNIX, Macintosh și Win95/NT.

CGI Made Really Easy

`http://www.jmarshall.com/easy/cgi/`

Este un ghid direct, fără banalități, care ridică vălul de pe misticismul și presupusa complexitate din Common Gateway Interface, cu exemple atât în limbajul Perl, cât și în limbajul C. Există legături spre alte informații CGI pe același sit sau în altă parte. Rulează pe platformele UNIX, Macintosh și Win95/NT.

CGI Resource Index

`http://www.cgi-resources.com/`

Conține peste 800 de legături diferite către resurse CGI, de la un alt furnizor de servicii – World Wide Mart. Resursele variază de la procesoare de formulare gata de folosire, la programatori pentru CGI profesioniști pe care puteți să-i angajați. Există o mulțime de scripturi pe acest sit, scrise în Perl și C++, iar multe dintre ele sunt cu adevărat utile. Este de asemenea gazda programului Matt's Script Archive (consultați următorul alineat). Rulează pe platformele UNIX, Macintosh și Win95/NT.

Script Archive de la Matt

`http://worldwidemart.com/scripts/`

Una dintre cele mai bune surse de scripturi CGI din lume, acest sit are câteva zeci de scripturi online în Perl și C, cu motoare de căutare, generatori de legături aleatoare, animație, ceasuri, verificatoare de cărți de credit și alte articole foarte folositoare. Una dintre cele mai bune modalități de a folosi astfel de situri, dacă nu puteți folosi chiar codul, este de a studia cum abordează un programator profesionist o anumită operație. Adesea puteți asimila destulă strategie pentru a fi în stare să abordați propria problemă cu mai multă încredere. Rulează pe platformele UNIX, Macintosh și Win95/NT.

Script Archive de la Selena Sol

`http://selena.mcp.com/Scripts/`

Selena Sol, alter ego-ul feminin al lui Eric Tachibana, are o colecție semnificativă de scripturi, cugetări, filozofii și resurse adunate aici și pe situri identice din toată lumea. De la parole de protecție pentru căutări în baze de date, până la cookie de prezicere a viitorului, situl Selenei Sol conține una dintre cele mai mari colecții de rutine CGI puse împreună vreodată, iar autorul merită să fie cunoscut prin paginile și munca sa. Rulează pe platformele UNIX, Macintosh și Win95/NT.

Perl (The Practical Extraction and Report Language)

Puteți folosi orice limbaj de calculator pentru a scrie un „script" CGI. De fapt, ele nu trebuie să fie chiar niște „scripturi". Puteți folosi limbajul COBOL, atât timp cât îl puteți face să se comporte și să discute așa cum dorește CGI. Majoritatea scripturilor CGI au fost scrise în limbajul Perl, un limbaj pentru „scriptarea" adevărată, special proiectat pentru a fi un limbaj de raportare pentru procesarea ușoară și rapidă a textului cu un efort minim de programare. A fost și este ideal pentru operație, deși mulți utilizatori folosesc limbajele Java, C/C++ sau alte limbaje specializate pe anumite operații. Java este interesant în special deoarece un distribuitor poate oferi un singur modul pentru o aplicație care va rula pe orice calculator. Tabelul G.15 rezumă câteva resurse și unelte specifice disponibile pe Web.

Tabelul G.15. Resursele Perl.

Resurse	*Tip/Caracteristici*	*Audiență*	*Gama de preț*	*Platforme*
Web Toolbox	Tutorial	Novice	Gratis	Oricare
Perl Language	Listă de resurse	Programatori	Gratis	Oricare
MKS Toolkit	Emulație UNIX Utilitare Editor Etc.	Programatori	$$$	Win95/NT MIPS, Alpha, PowerPC
Perl/Tk FAQ	Tutorial	Programatori	Gratis	Oricare
NCSA Perl	Tutorial	Programatori	Gratis	Oricare
Perl CGI	Exemple	Programatori	Gratis	Oricare

Web Engineer's Toolbox de la Jacqueline Hamilton

`http://lightsphere.com/dev/class/`

Conține mici lecții de limbaj Perl pentru utilizatorii Web începători. Presupun doar cunoștințe generale de HTML și probabil sunt cel mai bun punct de pornire, dacă nu cunoașteți deja un limbaj de programare. Nu depinde de anumite platforme.

The Perl Language Homepage

`http://www.perl.com/`

Este sursa centrală de informații despre limbajul Perl, un limbaj puternic, orientat pe obiect, proiectat special pentru text și manipulare lexicală. Deși aglomerat în comparație cu Visual Basic și alte limbaje de scriptare, există puține lucruri care nu pot fi făcute rapid și ușor în limbajul Perl, după ce ați început să învățați. Pentru programatorii care sunt deja familiarizați cu limbajele C sau C++, Perl este un limbaj ușor. Există multe legături și resurse aici, așa că fiți pregătiți să petreceți ceva timp pentru explorarea acestui sit. Rulează pe mai multe platforme fără dependențe deosebite.

Mortice Kern Systems (MKS) Toolkit

`http://www.mks.com/`

MKS Toolkit oferă utilitare și comenzi bazate pe sistemul UNIX, care facilitează tranziția în orice parte. Dacă cunoașteți mediul UNIX bine, Toolkit vă lasă să folosiți comenzile UNIX pe calculatoare Windows/DOS. Dacă cunoașteți sistemele DOS/Windows, MKS vă lasă să exersați acasă, înainte de a încerca lumea UNIX fără a avea o rețea sigură. Conține Perl, Awk, Vi, Tar, CPIO, Ksh și multe alte programe tip UNIX. Un utilitar opțional Pscript conține funcții Perlized ActiveX pentru Win95/NT. Rulează pe platforme Win95, WinNT și OS/2, inclusiv Intel, MIPS, Alpha și hardware PowerPC.

Perl/Tk FAQ

`http://w4.lns.cornell.edu/~pvhp/ptk/ptkTOC.html`

Acest sit conține informații despre limbajele fuzionate Perl și Tcl/Tk, un kit cu instrumente WIDGET de controale vizuale terminale. În principal, vă permite să adăugați o interfață de utilizator grafică la Perl, care este bazată pe text. Rulează în principal pe platforma UNIX X Window, dar au fost create porturi Win95/NT.

Perl Tutorial from NCSA

`http://www.ncsa.uiuc.edu/General/Training/PerlIntro/`

Conține o introducere simplă a tuturor funcțiilor de bază ale limbajului Perl, folosind o simplă funcție `mailto` ca instrument de învățare. La sfârșitul cursului pe care îl parcurgeți singur, ați obținut un instrument care funcționează.

Perl CGI Examples

`http://www.panix.com/~wizjd/test.html`

Conține programe CGI gratuite, destinate profesioniștilor care studiază interfața CGI. Rulează pe platformele UNIX, Macintosh și Win95/NT.

C/C++

De unde să începem? Limbajul C/C++ este o lume în sine, cu o cultură, limbaj, C-etățeni și clienți din toată lumea. C și C++ vă conferă acces direct la facilitățile complete ale oricărui calculator UNIX, pentru care reprezintă limbajul de programare preferat al sistemelor.

Dacă nu puteți rezolva ceva cu alte mijloace, C/C++ va funcționa garantat. Nu va fi chiar perfect, dar maeștrii C/C++ lucrează la asta. Tabelul G.16 prezintă câteva locuri bune pentru a începe contactarea locuitorilor lumii C.

Tabelul G.16. Resurse C/C++.

Resurse	*Tip/Caracteristici*	*Audiență*	*Gama de preț*	*Plaforme*
C++ Lite FAQ	Tutorial/FAQ	Programatori	Gratis	Oricare
Prog. Source	Listă de resurse Tutorial	Programatori	Gratis	Oricare
C Prog by Nerd	Listă de resurse Recenzii ale produselor Tutorial Referințe despre locuri de muncă	Programatori	Gratis	Oricare
Lysator C Prog	Listă de resurse Studii savante Tutorial	Programatori	Gratis	Oricare

C++ Lite FAQ

`http://www.cerfnet.com/~mpcline/c++-faq-lite/`

Un mic manual despre C++ și câteva legături reprezintă elementele majore cu care se mândrește, în afară de secțiunea FAQ (Frequently Asked Questions – întrebări frecvente), care este excelentă. Nu depinde de platforme.

The Programmer's Source

`http://www.progsource.com/`

Conține o listă bogată de legături la resurse, acoperind nu doar limbajul C/C++, ci și limbajele Java, CGI, Perl și alte lucruri interesante pentru programatori. Conținutul cuprinde referințe despre compilatoare, manuale, grupuri de discuție și arhive software. Nu depinde de platforme.

C Programming by Nerd World

`http://www.nerdworld.com/users/dstein/nw93.html`

Conține o colecție mică, dar semnificativă, de manuale și legături. Nu depinde de platforme.

Lysator C Programming

`http://www.lysator.liu.se/c/`

Conține o listă vastă de recomandări, unele dintre ele sunt vesele, dar majoritatea sunt într-adevăr foarte sincere. Dacă nu sunt aici, probabil că nu sunt importante. (Am glumit, așa că nu săriți la mine pentru asta). Nu depinde de platforme.

Visual Basic

Limbajul Visual Basic (VB) este limbajul ales de mulți programatori Windows/Win95. Are acces la resursele mediului Windows și există multe kituri de instrumente pentru controale disponibile pentru a face aproape tot ce doriți. În tabelul G.17 este prezentată o listă scurtă de resurse Visual Basic.

Tabelul G.17. Resurse Visual Basic.

Resurse	*Tip/Caracteristici*	*Audiență*	*Gama de preț*	*Plaforme*
MSDN	Resurse Articole de presă Tutorial	Programatori	Gratis	Win95/NT
MS Visual Basic	Resurse Articole de presă Tutorial	Programatori	Gratis	Win95/NT
IR for Win Developers	Resurse Recenzii Tutorial	Programatori	Gratis	Win95/NT
Visual Basic IR	Resurse Recenzii Tutorial	Programatori	Gratis	Win95/NT

The Microsoft Developers Network (MSDN)

`http://www.microsoft.com/msdn/`

Acest sit este o sursă principală de informații despre compilatoarele Microsoft, SKD-uri și alte instrumente pentru programatori. Dacă doriți să știți ce se pregătește în bucătăria Microsoft, aici este locul unde puteți să mirosiți aroma. Este dependent total de platforma Microsoft.

The Microsoft Visual Basic Site

`http://www.microsoft.com/vbasic/`

Este un sit puțin mai depărtat de MSDN și mult mai specific pentru proiectarea în Visual Basic. Prezintă pe larg unelte și tehnici pentru proiectanții VB. Depinde de platformele Microsoft.

Internet Resources for Windows Developers

`http://www.r2m.com/windev/`

Aceasta este o pagină de legături aflată pe un sit independent, Internet Resources for Windows Developers, care conține multe alte lucruri pe lângă Visual Basic și merită să fie explorat din alte motive. Depinde de platformele Microsoft.

Visual Basic Internet Resources

`http://www.r2m.com/windev/visual-basic.html`

Este un sit independent, bun pentru programatorii din Windows și este copilul sitului anterior care este mai cuprinzător. Puteți găsi de asemenea o mulțime de resurse pe siturile Microsoft Developers Network menționate anterior. Depinde de platformele Microsoft.

Firewall, servere proxy și securitate

Securitatea în Internet este un subiect de actualitate și o preocupare a tuturor în ziua de azi. Nu puteți deschide un ziar fără să auziți de date confidențiale obținute ilegal sau cum vreun infractor a atacat unele firme și chiar persoane fizice – numere de cărți de credit furate, identități furate, chiar programe secrete Java sau ActiveX care lasă telefonul conectat și apelează situri din altă parte a lumii pentru a vă încărca nota de plată la telefon, ca efect al refuzării unei companii de telefon străine fără scrupule.

Cum vă păstrați datele în siguranță? E destul de rău să vă deschideți corespondența și să vedeți că ați vorbit la telefon cinci ore cu Faroutistan, fără știrea dumneavoastră și în timp ce dormeați, dar dacă cineva pătrunde pe situl dumneavoastră și umple toate bazele de date cu date false sau fură informații strict-secrete referitoare la, să spunem, cea mai nouă și secretă linie de costume de baie, pentru a le vinde concurenței, atunci vorbim de ceva care ar putea distruge afaceri sau chiar vieți.

Programele tip firewall (parafoc) sunt răspunsul Internetului referitor la lacăte și porți ferecate, păstrând vizitatorii nedoriți afară și lăsând utilizatorii care știu parolele să intre și să iasă. Un server proxy este ca un fel de „șef al unui dormitor comun", care vă spune ce puteți să faceți și când puteți să faceți cerând ca toate apelurile „băieților" să fie afișate pe suprafața de lucru a calculatorului. Scopul lui este de a vă proteja de deciziile neînțelepte pe care le-ați putea lua, dacă aveți prea multă încredere în oameni (și diverse calculatoare suspecte).

În general, pachetele OS dedicate oferă cea mai bună siguranță și cea mai mică expunere, doar și fiindcă majoritatea programelor de atacare sunt disponibile pentru sistemele obișnuite, UNIX, WinNT etc. Pe de altă parte, ele au prețuri mari. Pentru Webmasterii care doresc să-și creeze propriile sisteme de siguranță, implementările proxy și firewall pe larg disponibile pentru UNIX oferă administratorului de sit Web cu multe resurse, posibilitatea de a-și crea un sistem de siguranță personal și ieftin. Tabelul G.18 prezintă lista resurselor pentru securitate disponibile pe Web.

Tabelul G.18. Firewall, servere proxy și securitate.

Produs	*Tip/Caracteristici*	*Audiență*	*Gama de preț*	*Plaforme*
WWW Sec. FAQ	Informații	Novice până la expert	Gratis	Oricare
I-Pad	Server sigur	Intermediar	$$$$	WinNT
Border Manager	Firewall	Avansat	$$$$$	Particulară
MS Proxy	Server proxy	Avansat	$$$$	WinNT
Netscape Proxy	Server proxy	Avansat	$$$$	WinNT/UNIX
Cisco PIX	Firewall	Avansat	$$$$$	Particulară
Checkpoint	Firewall	Avansat	$$$$$	WinNT/UNIX
SSLeay	Socluri sigure	Avansat	Gratis	UNIX
Apache SSL	Server sigur	Avansat	Gratis	UNIX

The WWW Security FAQ

`http:/www-genome.wi.mit.edu/WWW/faqs/www-security-faq.html`

Conține o introducere în securitatea pe WWW cu discuție de tip întrebare/răspuns pas cu pas, despre implicațiile siguranței pe World Wide Web.

I-Pad eSoft

`http://www.esoft.com/`

Este un pachet ce conține un server complet cu router TCP/IP și servere HTTP, FTP, DNS, SMTP, POP3 și Dial-up SLIP/PPP. Sunt incorporate câteva porți CGI, dar pachetele gata configurate nu pot fi adăugate. De asemenea, este încorporată și facilitatea de firewall. Este bun pentru situri mici și medii care necesită urgent un sistem simplu. Platforma Intel WinNT face parte din pachet.

Novell Border Manager

`http://www.novell.com/`

Este un firewall certificat de NCSA, pentru sisteme medii și mari. Necesită o investiție minimă în obținerea licenței, prin administrare centralizată și un cod curat, bine optimizat. Rulează pe platformele dedicate IntranetWare cu Intel.

Microsoft Proxy

`http://www.microsoft.com/proxy/ http://www.micorsoft.com/`

Este un server proxy orientat pe partea de server, ușor de folosit și relativ puternic pentru situri mici și medii, care folosește sistemele de operare Microsoft inițiale, dar are și scalabilitatea, standardizarea și problemele de siguranță ale sistemului WinNT Server când se încearcă extinderea la o organizație mai mare. Rulează pe platforma WinNT Server.

Netscape Proxy Server

`http://www.netscape.com/comprod/netscape_products.html http://www.netscape.com/`

Acesta este un server proxy pe larg folosit pentru platforme obișnuite, care necesită mai puține cunoștințe și mai puțină îndemnare pentru a configura un sistem decent, decât variantele UNIX de tip „configurează-l singur". Se integrează bine cu Netscape Commerce Server și alte servere Netscape. Rulează pe WinNT Server, Alpha Server și platformele UNIX.

Cisco Systems PIX

`http://www.cisco/pix/`

Este o soluție flexibilă de firewall bazat pe hardware, certificată de NCSA, pentru firmele medii și mari. Necesită o investiție minimă pentru licență. Rulează pe platformele OS înglobate dedicate.

Check Point Firewall

`http://www.checkpoint.com/`

Dacă deține două cincimi din piața mondială de programe firewall, înseamnă că Check Point face ceva treabă bună. Este un firewall și un sistem de siguranță complet echipat, bazat doar pe software, oferind o gamă completă de facilități și opțiuni pentru Webmasterii pretențioși. Rulează pe platformele UNIX și WinNT.

SSLeay și SSLapps

`http://www.psy.uq.oz.au/~ftp/Crypto/`
`http://apache.org/`
`http://www.algroup.co.uk/Apache-SSL/`

Secure Sockets Layer pentru platformele UNIX și WinNT. Combinația de SSLeay și Apache SSL face posibilă rularea unui server sigur „făcut în casă", pe aproape orice calculator. S-ar putea să fie necesar să cumpărați o licență de sit dintr-o sursă verificată (cum ar fi Verisign, de exemplu), dar până în momentul în care citiți această carte problema poate fi deja rezolvată de o comunitate activă internațională de dezvoltare. Este disponibil codul sursă.

Apache SSL

`http://www.apache.org/`
`http://www.algroup.co.uk/Apache-SSL/`

Este o modificare a serverului Apache prezentat la secțiunea anterioară „Servere Web". Consultați discuțiile și corecturile aduse codului Apache la a doua adresă URL. Serverul

însuși este disponibil la prima adresă URL. Alte discuții interesante le puteți găsi la secțiunea „Criptografie internațională" din acest capitol.

Biblioteci și alte resurse pe Web

Acum, încotro mai mergeți de aici? Această carte v-a spus o mulțime de lucruri despre HTML, iar această anexă a acoperit o mulțime de chestiuni. Dar există destule lucruri pe care nu am avut timp sau spațiu să le menționăm, chiar și într-o atfel de carte imensă, iar lucrurile se schimbă în continuu. Din fericire, există situri dedicate, pentru păstrarea informațiilor actualizate la zi, 24 de ore din 24, șapte zile pe săptămână. Unele dintre ele sunt lucrări voluntare ale unor oameni care au fost interesați de un anumit subiect, iar unele sunt situri comerciale care publică echivalentul pe Web al unei reviste, permițându-vă accesul la articole și informații în schimbul faptului că citiți reclamele pe care ei vi le prezintă. Tabelul G.19 prezintă câteva referințe la unele gazde de informații pe care nu le găsiți ușor în altă parte.

Tabelul G.19. Resurse Web.

Resursa	*Tip*	*Audiență*	*Gama de preț*	*Platforme*
W3C	Organizație de standarde	Întregul Web	Nedisponibil	Oricare
HTML	Definiții DTD în HTML	Programatori	Nedisponibil	Oricare
CSS	Foi de stiluri	Programatori	Nedisponibil	Oricare
DOM	Model obiect	Programatori	Nedisponibil	Oricare
Math	Marcator matematic	Programatori	Nedisponibil	Oricare
Graphics	Standarde GFX	Programatori	Nedisponibil	Oricare
Access	Accesibilitate	Programatori	Nedisponibil	Oricare
XML	Extindere HTML	Programatori	Nedisponibil	Oricare
NS DevEdge	Resurse NS	Programatori	Gratis până la $$$$	Oricare
Site Builder	Resurse MS	Programatori	Gratis până la $$$$	MS
Web Reference	Bibliotecă	Programatori	Gratis	Oricare
Macmillan	Bibliotecă	Programatori	Gratis	Oricare
HTML Guru	Bibliotecă	Programatori	Gratis	Oricare
HTML WG	Corporație/ Bibliotecă	Programatori	Gratis	Oricare
WDVL	Bibliotecă	Programatori	Gratis	Oricare
Stars.com	Cionă WDVL	Programatori	Gratis	Oricare
Webmaster	Bibliotecă	Programatori	Gratis	Oricare
WebABLE!	Listă de resurse	Oricine	Gratis	Oricare
CAST	Listă de resurse	Oricine	Gratis	Oricare
Low Vision	Listă de resurse	Oricine	Gratis	Oricare

(continuare)

Tabelul G.19. - continuare

Resursa	*Tip*	*Audiență*	*Gama de preț*	*Platforme*
Blindness	Listă de resurse	Oricine	Gratis	Oricare
Cryptography	Listă de resurse	Programatori	Gratis	Oricare
Spiderwoman	Listă poștală	Programatori	Gratis	Oricare
VBScript	Listă poștală	Programatori	Gratis	MS
ActiveX	Listă poștală	Programatori	Gratis	MS
CGI List	Listă poștală	Programatori	Gratis	Oricare
Advanced CGI	Listă poștală	Programatori	Gratis	Oricare

World Wide Web Consortium (W3C)

`http://w3c.org/`

Este sursa centrală de informații despre ce se întâmplă în lumea adevărată, spre deosebire de „serele" creatorilor principali de browsere. Deși Microsoft și Netscape domină o bună parte a datelor prelucrate de acest for care promulgă standarde, doar fiindcă își permit să plătească specialiști pentru a lucra la mai multe proiecte, ambele firme sunt încă înhămate la o muncă grea timp îndelungat. De aici vin noile standarde și aici se vor muri vechile standarde. W3C realizează și câteva teste și eforturi de cercetare, inclusiv browserul de probă Amaya, cât și serverul bazat pe Java, Jigsaw. Nu depinde de platforme.

HTML 3.2 și 4.0

`http://w3.org/MarkUp/`

Aceasta este pagina care ar trebui să fie marcată deoarece subpaginile se mută sau chiar dispar. Conține referințe către toate documentele W3C curente păstrate pe sit, cât și multe legături către sursele exterioare de informații, o caracteristică specifică paginilor W3C. Nu depinde de platforme.

Cascading Syle Sheets (CSS)

`http://w3c.org/Style/`

Acesta este rădăcina arborelui pentru foile de stil, care conține specificațiile curente, cât și legături spre documentele de lucru pentru Dynamic HTML și Cascading Style Sheets. Acesta este de asemenea un punct de pornire pentru documentul referitor la standardul ISO pentru foile de stil, care este teoretic mai avansat (și complex): DSSL (Document Style Semantics and Specification Language).

Document Object Model (DOM)

`http://w3c.org/MarkUp/DOM/`

Referitor la Dynamic HTML, iată punctul de pornire al consorțiului W3C pentru DOM Working Group – grupul de lucru DOM, care încearcă să împace ideile incompatibile (în principal) ale companiilor Microsoft și Netscape și să asigure că orice program scris în Dynamic HTML va funcționa fără să creeze probleme altor utilizatori. Interesant dacă va funcționa! Se pare că o versiune a arhitecturii Microsoft va câștiga, dar nu se știe niciodată; HTML 3.0 a fost pentru un timp un produs important, dar nimeni nu se mai laudă cu el azi. Nu depinde de platforme.

Math on the Web

`http://www.w3.org/Math/`

Și dacă tot am vorbit de HTML 3.0, iată locul unde faimoasele elemente matematice din HTML 3.0 au suferit modificări importante. Acesta este centrul Grupului de lucru HTML Math - HTML Math Working Group – și locul unde puteți să mergeți dacă doriți vreodată să publicați matrice monodromice geometrice sau alte ezoterii, pentru un auditoriu internațional, așteptând fără îndoială, cu sufletul la gură o modalitate mai ușoară de a vedea aceste revelații sublime. Nu prea avem nici o idee ce este monodromia, așa că vă rugăm nu ne întrebați, dar este probabil apogeul a miliarde de ani de evoluție și formează baza ideii „Viața așa cum o știm". Nu depinde de platforme.

Graphics on the Web

`http://w3c.org/Graphics/`

Conține tot ceea ce ați dorit vreodată să știți despre formatele de fișiere grafice, dar v-a fost teamă să întrebați. GIF, TIFF, JPEG, PNG, PostScript, CGM – toate se găsesc aici (sau cel puțin legături către ele) cu discuții despre care va fi rolul lor pe Web în viitor. Ultimele idei spun că formatele GIF sunt pe cale de a fi înlăturate din cauza insistenței actuale a celor de la Unisys de a se aplica drepturile de autor (discutabile) referitor la compresia LZW, pe care se bazează formatul GIF. PNG (Portable Network Graphics), înlocuitorul preferat, este încă în faza de proiectare și pe cale de a fi propagat pe Web. Nici una dintre referințele de pe sit nu sunt foarte noi, așa că se pare că ideea s-a cam pierdut. Evident că dacă formatul PNG va înlocui formatul GIF, patentul Unisys nu prea va avea nici o valoare, poate doar pentru istoricii rețelei Web. Pe de altă parte, înlocuirea tuturor fișierelor GIF care se află pe Web ar putea fi o muncă uriașă, așa că probabil deranjul merită suma cerută de Unisys. Ceea ce e sigur e că dacă sunteți un proiectant Web, ar trebui să fiți mai atent ce formate grafice folosiți, deoarece ar putea să se întoarcă și să vă urmărească dacă administrați un sit comercial. Nu depinde de platforme.

The Web Accessibility Initiative

`http://w3c.org/WAI/`

Acesta este situl Accesibility Central, unde ei țin cont de milioanele de oameni din lume cu handicapuri temporare sau permanente. Pagina Reference – referință, este un excelent loc de pornire pentru căutare. Se pare că accesibilitatea a devenit în final o parte reală din standardele HTML, așa cum WAI are o anumită influență asupra anumitor caracteristici din HTML 4.0, dar încă mai sunt multe de făcut. Acesta este un bun loc de a găsi situri care conțin accesibilitate,cât și browsere specializate, adesea folosite de persoanele cu handicapuri deși această anexă vă oferă o bună prezentare.

Extensible Markup Language (XML)

`http://w3c.org/XML/`

Aici puteți găsi legături către informații despre limbajul XML (Extensible Markup Language) care vă vor permite să proiectați propriile definiții DTD, etichete și atribute. Ei pretind că aceasta va face documentele mai lizibile, dar eu nu prea cred, deoarece doar câțiva programatori pot să utilizeze setul de etichete care formează standardele HTML existente. Pe de altă parte, XML este cel mai mare lucru de la inventarea „feliei de pâine care cade întotdeauna cu partea unsă cu unt în sus", conform spuselor specialiștilor. Nu depinde de platforme.

Other W3C Issues

`http://w3c.org/`

Alte elemente lansate de W3C sunt browserul de test Amaya și serverul HTTP Jigsaw menționat anterior, internaționalizarea (în special setul de caracteristici pentru font și caracter), fonturi (propagarea fonturilor pe Web, dar și internaționalizarea), Digital Signature Initiative (inițiativa semnăturii digitale), comerțul electronic, metadatele (o activitate generalizată pentru a stabili cum să vă exprimați opiniile – afirmațiile, mai degrabă – despre identitatea indivizilor, conținutul siturilor Web și crearea distribuită și mecanismele de lansare a versiunilor), PICS (sisteme de evaluare a conținutului), drepturi de autor, confidențilitate și preferințe, securitate, HTTP 1.1., tehnologia orientată pe obiect (CORBA, DCE, Microsoft COM, Plan 9), audio, video și multimedia sincronizată. Toate aceste elemente pot fi găsite cu ajutorul legăturilor din pagina gazdă a acestui sit.

Netscape DevEdge

`http://developer.netscape.com/`

Este salonul programatorilor de la Netscape, pentru aceia care au un orizont Netscape. Conține multe legături către resurse interne și externe. Are niveluri multiple de apartenență. Nu depinde de anumite platforme – deși nu pare a avea prea multe legături către Microsoft!

Microsoft Site Builder Network

`http://www.microsoft.com/sitebuilder/`

Este un sit bazat pe platforma Microsoft, care conține tone de informații valoroase despre instrumentele și produsele Microsoft. Dacă trăiți într-o lume Microsoft, acesta este locul unde puteți găsi tot ce e posibil de găsit, dar orizontul se îngustează atunci când priviți dincolo de zidurile virtuale. Permite niveluri multiple de apartenență. Rulează pe platformele Microsoft (Win95/NT).

Web Reference

`http://www.webreference.com/`

Este o bună introducere generală într-o mulțime de subiecte și manuale despre tehnologiile uzuale, cum ar fi JavaScript.

Macmillan's HTML Workshop and Survival Guide

`http://www.mcp.com/general/workshop/`

Conține manuale foarte bune și o introducere în HTML, oferite de faimoasa editură care a publicat multe lucrări de referință în domeniul calculatoarelor, lucrări de o valoare nemăsurată, în special cartea de față. Nu depinde de platforme.

Chuck Musciano's The HTML Guru

`http://members.aol.com/htmlguru/`

Conține o scurtă colecție de sfaturi și trucuri oferite de un autor individual. Unele dintre avantajele acestui sit, recomandat în special începătorilor, sunt spațiul pentru date, uniform și bine organizat, unde puteți căuta și modelul de întrebare/răspuns pentru o mare parte din acest material. Cu toate că există doar un singur punct de vedere, puteți echilibra părerile din acest sit cu altele. Nu depinde de platforme.

The HTML Writer's Guild

`http://hwg.org/`

Acest sit este locul principalelor organizații pentru profesioniștii Web, cât și o bună sursă de informații despre toate aspectele profesiei, locuri de muncă, forumuri și alte aspecte folositoare. Sunt disponibile diverse niveluri de apartenență. Nu depinde de platforme.

Web Developer's Virtual Library (WDVL)

`http://www.wdvl.com/`

Este o resursă excelentă și la zi pentru oricine care este implicat în proiectarea sau întreținerea siturilor Web. Conține o prezentare foarte bună a limbajului HTML 4.0 cu exemple de manuale și legături către multe alte resurse, inclusiv multe dintre specificațiile de la W3C și din alte surse. Este prezentat aproape orice subiect la care vă puteți gândi, într-un index foarte bine organizat pe pagina de deschidere, așa că este extrem de simplu să căutați ceea ce vă interesează. Nu depinde de platforme.

The Virtual Library of WWW Development

`http://www.stars.com/Vlib/`

Conține o altă versiune de WDVL discutată anterior.

Webmaster Resources

`http://www.cio.com/resources/`

Conține un compendiu bine organizat de resurse și unelte pentru webmasteri, inclusiv dezbateri despre profesie, societăți profesionale, manuale, HTML, grafică, situri de publicitate, motoare de căutare, tehnologii și alte subiecte interesante pentru profesioniștii în Web. Nu depinde de platforme.

Yuri Rubinsky Insight Foundation WebABLE!

`http://www.yuri.org/webable/`

Într-un anumit moment din viața dumneavoastră, există posibilitatea să suferiți de un handicap, poate temporar poate nu chiar atât de temporar. Acest sit conține o colecție excelentă de legături și reguli de proiectare pentru siturile Web inclusive, pentru a susține utilizatorii cu diverse tipuri de incapacități. Am discutat despre aceasta anterior, la validatorul HTML Bobby, dar ne putem repeta. Acum că W3C a făcut posibilă codarea accesibilității, este doar o problemă de timp înainte de a fi introdusă într-un proiect profesionist. Acum e un moment excelent pentru a studia articolele implicate înainte de a deveni literă de lege. Nu depinde de platforme.

CAST – Center for Applied Special Technology

`http://www.cast.org/`

Validator de accesibilitate și gazda validatorului Bobby HTML, situl CAST este de asemenea o resursă excelentă pentru regulile de proiectare și legături spre alte organizații, cât și articole referitoare la proiectarea inclusivă. Nu depinde de platforme.

Low Vision and Deaf/Blind

`http://home.earthlink.net/~robinpow/blind.htm`

Este o listă de resurse excelentă și extensivă pentru utilizatorii care au deficiențe de auz și văz. Nu depinde de platforme și este foarte compatibil cu Lynx.

Blindness Resource Center

http://www.nyise.org/speech/blind.htm

Este o listă de legături excelentă pentru utilizatorii nevăzători, având o structură mai ierarhizată decât precedentul sit Low Vision and Deaf/Blind. Foarte bine acoperit de distribuitorii comerciali, care oferă produse pentru utilizatori nevăzători și legături pentru toți cei interesați, inclusiv o bună resursă despre „Istoria scrierii codurilor pentru nevăzători", care este alterată doar de folosirea intensivă a graficii (deși este greu de imaginat cum pot fi reprezentate în caractere ASCII, versiunile mai vechi de Braille, New York Point și sistemele de litere Roman în relief). Poate că un text alternativ va ajuta. Este disponibilă o versiune în alb și negru. Nu depinde de platforme.

International Cryptography

http://www.cs.hut.fi/crypto/

Este o resursă vastă despre criptografie și securitate, din Finlanda, care se concentrează pe un software disponibil în afara Statelor Unite, pentru a evita resptricțiile impuse la exportul de software din SUA. Conține referințe bune de manuale și permite accesul la toate informațiile necesare pentru a configura un SSH (Secure Shell), SSLeay/Apache-SSL, portofele electronice și alte aplicații sigure. Rețineți că tehnicile și produsele descrise aici s-ar putea să nu fie legale sau poate nu este legal să le copiați în țara dumneavoastră. Rulează pe mai multe platforme, dar mare parte din freeware este sub UNIX.

Stephanie Brail's Spiderwoman Mailing List at Amazon City

http://www.amazoncity.com/spiderwoman/

Este o resursă excelentă pentru femeile care lucrează în domeniul proiectării pe Web, cu sfaturi, resurse, o cameră de discuții și alte materiale interesante în special pentru femei – și, bineînțeles, excelenta listă însăși. Nu depinde de platforme.

VBScript Mailing List

Este o resursă excelentă pentru programatorii în VBScript pe platformele Microsoft. Acesta este un loc unde se adună maeștrii grei din lumea VBScript, răspund la întrebări și își împărtășesc experiența. Rulează pe platformele Microsoft.

Pentru a vă înscrie, trimiteți un e-mail la adresa: `Listserv@listserv.msn.com`

Iar în corpul mesajului scrieți:

```
Subscribe VBScript Nume Prenume
```

ActiveX Controls Mailing List

O resursă similară pentru programatorii care scriu sau folosesc controale ActiveX pe platforme Microsoft. Aici puteți găsi răspunsuri la probleme controversate, sfaturi sau chiar vă puteți împărtăși cunoștințele. Rulează pe platformele Microsoft.

Pentru a subscrie, trimiteți un e-mail la adresa: `Listserv@listserv.msn.com`

Iar în corpul mesajului scrieți:

```
Subscribe ActiveXControls Nume Prenume
```


Common Gateway Interface List

Conține o listă pentru programatorii în CGI pentru a pune întrebări și a-și împărtăși experiența despre scrierea codurilor pentru această interfață populară. Nu depinde de platforme.

Pentru a subscrie, trimiteți un e-mail la adresa: `listserv@vm.ege.edu.tr`

Iar în corpul mesajului scrieți:

```
Subscribe CGI-L Nume Prenume
```

Advanced CGI Discussion List

Conține dezbateri și întrebări despre Common Gateway Interface și cum se creează programe sau scripturi care accesează această interfață. Nu depinde de platforme.

Pentru a subscrie, trimiteți un e-mail la adresa: `listproc@lists.nyu.edu`

Iar în corpul mesajului scrieți:

```
Subscribe ADV-CGI Nume Prenume
```

Biblioteci și alte resurse: în principal HTML

Există atât de multe resurse extraordinare disponibile pe Web, că este greu de știut la care să vă opriți. La sugestia unor partizani declarați ai siturilor Web, am inclus încă o secțiune cu diverse resurse pentru a nu fi geloși pe listele de resurse cu mai multe cifre, dar mai puține esență. La fel ca acele persoane care vând seturi de cuțite „Shogun" prin emisiuni TV am inclus aici nu una ci două liste mari de resurse: lista generală comodă, din secțiune anterioară „Biblioteci și alte resurse pe Web" și aceasta, care se concentrează pe HTML.

HTML este un subiect vast, în ciuda faptului că a fost puțin umbrit de multiplele extensii multimedia, care par să apară uneori mai repede decât putem face față și, de aceea merită să ne oprim aici pentru a avea în vedere că de aici a pornit totul și, fără a înflori, HTML este încă apt să prezinte o pagină interesantă și informativă. Am inclus și câteva referințe interesante către paginile cu istorie, dorind să fac această listă utilă ca resursă cultural pentru cetățenii sofisticați ai Internetului, cât și un ghid pentru utilizatorii HTML.

Tabelul G.20 prezintă un rezumat al resurselor HTML în principal, dar este o referință la fel de bună ca și secțiunea precedentă.

Tabelul G.20. Resurse HTML.

Resursa	*Tip*	*Audiență*	*Gama de preț*	*Platforme*
As We Think	Eseu	Toți utilizatorii	Gratis	Oricare
Xanadu	Eseu/Proiect	Toți utilizatorii	Gratis	Oricare
Berners-Lee	Conversații	Toți utilizatorii	Gratis	Oricare
WebTools	Tutorial	Programatori	Gratis	Oricare
Compendium of HTML	Index	Programatori	Gratis	Oricare
Ultimate Tag Reference	Index	Programatori	Gratis	Oricare
HTML 4.0 DTD	Documentație	Programatori	Gratis	Oricare
HTML Station	Tutorial	Programatori	Gratis	Oricare
DOT.html	Tutorial	Programatori	Gratis	Oricare
Netcaster	Tutorial	Programatori	Gratis	Oricare

(continuă)

Tabelul G.20. – continuare

Resursa	*Tip*	*Audiență*	*Gama de preț*	*Platforme*
Style Guide	Ghid	Programatori	Gratis	Oricare
Web Help	Tutorial	Programatori	Gratis	Oricare
Yale Guide	Tutorial	Programatori	Gratis	Oricare
Joe Barta	Tutorial	Programatori	Gratis	Oricare
Guide to Publishing	Tutorial	Programatori	Gratis	Oricare
HTML Card	Card de referință	Programatori	Gratis	Oricare
Jan's Guide	Tutorial	Programatori	Gratis	Oricare
HTML by Example	Tutorial	Programatori	Gratis	Oricare
Netscape Cookies	Tutorial	Programatori	Gratis	Oricare
Malcolm's Cookies	Tutorial	Programatori	Gratis	Oricare
Andy's Cookies	Tutorial	Programatori	Gratis	Oricare
Advanced HTML	Listă poștală	Programatori	Gratis	Oricare
Newsgroups	Știri	Programatori	Gratis	Oricare

As We May Think

`http://www.isg.sfu.ca/~duchier/misc/vbush/`

`http://www.ausbcomp.com/~bbott/wik/vbush.htm`

Acesta este articolul scris de Vannevar Bush, despre care mulți cred că a fost o previziune a ceea ce este World Wide Web azi. Este scris din perspectiva anului 1945, imediat după al doilea război mondial, totuși este plin de optimismul acelor ani când mulți cetățeni americani s-au simțit ca și cum ar fi ieșit dintr-un lung tunel întunecat și au pășit în lumina soarelui. Este un apel și un îndemn pentru reunirea „oamenilor de știință" (ca să nu fim prea nostalgici cu privire la acei ani, ei chiar erau oameni de știință) pentru a face accesibile pentru oricine cunoștințele dobândite și invențiile create în timpul acelor ani de război, prin folosirea noilor tehnologii care nu fuseseră inventate încă, dar el le-a putut prevedea. Nu depinde de platforme.

Ted Nelson's Xanadu

`http://www.xanadu.net/xanadu/`
`http://jefferson.village.virginia.edu/elab/hf10155.html`
`http://www.wired.com/wired/3.06/features/xanadu.html`
`http://www.xanadu.net/ararat`
`http://www.w3.org/Xanadu.html`

Acest sit conține o mică parte din povestea lui Xanadu, biblioteca hipertext de concepte și documente ale celui care a inventat cuvântul *hipertext* și este autorul multor idei pe care alți oameni le-au împrumutat sau reinterpretat mai târziu pentru a crea rețeaua Web așa cum o știm noi azi. La fel ca mulți alți vizionari înzestrați, Theodor Holm Nelson a întâmpinat dificultăți uneori când încerca să comunice cu restul oamenilor, așa că era nevoie de multă interpretare pentru ca oamenii obișnuiți să înțeleagă ce voia să spună. Acum există o multitudine de inovații ale căror origini se află în Xanadu, dar ai căror strămoși sunt necunoscuți. Nu depine de platforme.

Tim Berners-Lee

`http://www.w3.org/Talks/9510_Bush/Talk.html`
`http://www.taponline.com/rant/download/berners/page1.html`
`http://www-unix.oit.umass.edu/~abhu000/rp593m.disc.tbl.html`

Nici o discuție despre istoria rețelei Web nu este completă fără menționarea lui Tim Berners-Lee, care a adunat împreună toate ideile teoretice într-un server și browser HTTP din 1990 până în 1991 care funcționa. În sfârșit! El este nu numai o persoană inteligentă și practică, dar are și unele viziuni surprinzătoare despre ce a făcut lumea din creațiile lui. Ultima referință Web este un proiect de colaborare interesant care exemplifică puterea rețelei Web pentru unele tipuri de căutări. Încercați. Nu depinde de platforme.

WebTools.org

`http://www.webtools.org`

`http://www.webtools.org/counter/`

Conține o colecție interesantă de resurse generale HTML, inclusiv scriptul WebTools.org HTML Access Counter. Nu depinde de platforme.

Compendium of HTML Elements

`http://www.htmlcompendium.org/sitemap.htm`

Această listă completă a tuturor etichetelor HTML folosite, conține totul, de la etichete speciale WebTV, până la cele mai noi elemente HTML 4.0. Compendiul conține toate atributele pentru fiecare etichetă, cât și exemple specifice de folosire pentru fiecare element și atribut. Puteți să le descărcați pentru folosirea pe propriul sistem sau puteți folosi versiunea online pentru a vă asigura că aveți ultimele actualizări. Nu depinde de platforme.

D.J. Quad's Ultimate Tag Reference List

`http://www.quadzilla.com/reference/reference.htm`

Acest sit conține toate elementele organizate în pachete cu referințe clare, fiecare conținând un exemplu de folosire în cod. Nu depinde de platforme.

Guide to web Style

`http://www.sun.com/styleguide/`

Această referință comodă de la Sun Microsystems (programatorii limbajului Java) acoperă totul, de la alegerea scopului sitului dumneavoastră și audiența căreia se adresează, până la folosirea graficii. Nu depinde de platforme.

HTML 4.0 Document Type Definition HTML Version

`http://www.webtechs.com/sgml/HTML4/`

Acest sit de serviciu public de la WebTechs conține fiecare element din noua specificație HTML elaborată de W3C, într-un format ușor accesibil. Nu depinde de platforme.

HTML Station

`http://www.december.com/html/`

Este o resursă completă de referințe pentru un programator Web. Acoperă toate aspectele limbajului HTML, de la versiunea 0 la versiunea 4.0. Conține secțiuni despre CGI, formulare, tabele, seturi de caractere, extensii Netscape și Internet Explorer și surprize. Dar partea cea mai importantă este un manual de proiectare Web. Nu depinde de platforme.

Index DOT HTML

`http://www.blooberry.com/html/index.html`

Disponibil atât în versiunile online, cât și de descărcare, acest sit conține o istorie completă a versiunilor Hypertext Markup Language, Netscape și Internet Explorer. Eticheta History prezintă exact ceea ce era disponibil în fiecare versiune. În plus față de această achiziție considerabilă, situl se laudă cu o etichetă Tree care prezintă explicit relația de container dintre elementele HTML. Exemplele detaliate ale fiecărui element prezintă cum este folosit, pe ce browsere funcționează (și alte probleme pe care browserele le au când lucrează cu acest element) și aproape tot ceea ce doriți să știți. Și dacă nu este destul, există și un manual despre Cascading Style Sheets. Nu depinde de platforme.

Netcaster Developers Guide

`http://developer.netscape.com/library/documentation/netcast/devguide/index.html`

Este ghidul oficial pentru realizarea transmisiilor cu Netscape Navigator. Nu depinde de platforme, dar este orientat spre browserul Netscape.

Style Guide for Online Hypertext

`http://www.w3.org/Provider/Style/Overview.html`

Deși e puțin cam depășit, tema acestei lucrări este încă de actualitate. Tim Berners-Lee, arhitect principal al rețelei World Wide Web, își prezintă părerile despre buna proiectare a unui sit Web. Nu depinde de platforme.

Web Designer Help Desk

`http://web.canlink.com/helpdesk/`

Acest sit conține secțiuni despre crearea paginilor HTML, proiectarea siturilor Web, tabele, cadre, Java, motoare de căutare și etica Internet (Netiquette). Nu depinde de platforme.

Yale C/AIM Web Style Guide

`http://info.med.yale.edu/caim/manual/`

Reprezintă una dintre cele mai importante, dar mai puțin prezentate, zone ale proiectării paginilor web. Acest sit vorbește despre machetare, tipografie, multimedia, lucrări. Necesită un browser compatibil JavScript pentru a viziona sau descărca întregul ghid sub forma unui fișier Adobe PDF.

`ftp://info.med.yale.edu/pub/caim/pdf/s_guide.pdf`

Nu depinde de platforme.

Joe Barta's Form, Frame, Table and HTML Tutorials

`http://junior.apk.net/~jbarta/tutor/forms/index.html`
`http://junior.apk.net/~jbarta/tutor/frames/index.html`
`http://junior.apk.net/~jbarta/tutor/tables/index.html`
`http://junior.apk.net/~jbarta/tutor/makapage/index.html`

Este extraordinarul sit al lui Joe Barta cu manuale pentru crearea paginilor Web, având câte ceva de oferit pentru aproape oricine care lucrează cu limbajul HTML. Indiferent dacă sunteți doar la început sau căutați o sursă bună de sfaturi și trucuri, Joe Barta a pus pe sit, într-un format concis și clar, aproape tot ceea ce ați dori să știți despre folosirea formularelor, cadrelor, tabelelor și proiectarea generală a paginilor Web. Folosiți versiunea online sau le puteți descărca. Nu depinde de platforme.

Guide to Publishing HTML and Forms

`http://www.2kweb.net/guide-to-publishing-html.html`

Este un manual excelent cu prezentări vizuale ale modului cum diversele valori ale atributelor afectează imaginea unei pagini. Este un sit în dezvoltare, iar secțiunile viitoare vor acoperi și implementarea fișierelor MIDI, atât în Netscape, cât și în Internet Explorer. Nu depinde de platforme.

HTML Quick Reference

`http://sdcc8.ucsd.edu/%7Emiwilson/htmlref.html`

Deși nu este încă actualizat pentru HTML 4.0, constituie un excelent card de referințe. Nu depinde de platforme.

Jan's Guide to HTML

`http://www.ftech.net/~dutch/guide/`

Este un ghid complet al limbajului, conținând inclusiv ceva JavaScript, formulare, tabele și o prezentare de software pentru programatorii Web. Nu depinde de platforme.

Learning HTML by Example

`http://jeffco.k12.co.us/high/awest/learnhtml/00conten.htm`

Este unul dintre cele mai bine proiectate manuale HTML. Nu depinde de platforme.

Netscape's Persistent Cookies Documentation

`http://home.netscape.com/newsref/std/cookie_spec.html`

Conține documentație Netscape despre cum să folosiți elementele cookie pentru a salva informația de la o sesiune la alta sau în timpul unei tranzacții. Nu depinde de platforme.

Malcolm's Guide to Persistent Cookies

`http://www.emf.net/~mal/cookiesinfo.html`

Conține informații similare, dar dintr-o sursă diferită. Nu depinde de platforme.

Andy's Netscape HTTP Cookie Notes

`http://www.illuminatus.com/cookie`

Conține informații similar, dar dintr-o sursă diferită. Nu depinde de platforme.

Advanced HTML Discussion List

Este o sursă excelentă de sfaturi despre limbajul HTML, problemele pe care le întâmpină programatorii din Web și depășirea lor în crearea unui sit Web, cât și alte subiecte interesante pentru breasla programatorilor de pagini Web. Nu depinde de platforme.

Pentru a vă înscrie, trimiteți un mesaj e-mail la adresa: `listserv@ua1vm.ua.edu`.

În mesaj scrieți:

```
Subscribe ADV-HTML Nume Prenume
```

Grupuri de discuție

Iată câteva dintre sutele sau chiar miile de grupuri de discuții care conțin articole interesante pentru programatorii din Web. Cea mai bună modalitate de a descoperi care grup vi se potrivește este de a urma o legătură de la un sit sau o recomandare de la un prieten. Eu nu folosesc grupurile de discuții deoarece am remarcat că primesc prea multe comentarii

străine de subiect de la oameni neserioși. Eu prefer mediul mult mai controlabil al unei liste poștale, unde oameniii care creează probleme pot fi dați afară ușor. Această măsură de control tinde să crească nivelul general al expunerii. Dar mulți oameni iubesc grupurile de discuții și citesc o mulțime în fiecare zi.

```
comp.infosystems.www.authoring.html
comp.infosystems.www.authoring.cgi
comp.infosystems.www.authoring.misc
comp.infosystems.www.browsers.mac
comp.infosystems.www.browsers.ms-windows
comp.infosystems.www.browsers.x
comp.infosystems.www.browsers.misc
```

Observații pentru studii viitoare

Aceasta este doar o introducere în multitudinea de informații disponibile gratuit pe Web, cât și un punct de pornire pentru propriile dumneavoastră studii. Bineînțeles, nici o listă nu poate fi completă, așa că un anumit program sau resursă, ce poate fi chiar favoritul dumneavoastră necuprins în această listă, nu reflectă calitatea lui și nici toate resursele pomenite aici nu sunt perfecte. Ceea ce este adevărat este că fiecare dintre programele și utilitarele menționate are un auditoriu pentru care acesta este ideal sau cel puțin este potrivit. Am încercat să vă spunem suficient despre fiecare, așa că puteți decide singuri înainte de a hoinări pe Web pentru a găsi altceva ce vă poate servi în scopuri nepământești.

Când trebuie să hoinăriți, totuși, există strategii și tehnici pe care le puteți folosi pentru a vă asigura că munca dumneavoastră nu este în van. Lista de legături și bibliotecile listate în această anexă reprezintă puncte de pornire excelente pentru o căutare individuală specifică, dar uneori doriți să cunoașteți câte ceva despre un subiect de care nu a auzit nimeni, din câte știți.

Aici intră în scenă motoarele de căutare și păianjenii Web. Există două tipuri principale de căutare, aceea bazată pe căutarea după cuvinte cheie și aceea bazată pe organizarea după tematică. Un exemplu de sit cu cuvinte cheie este Alta Vista, iar Yahoo! este probabil cel mai cunoscut sit tematic. Totuși, apare o mică confuzie deoarece Yahoo! permite pe situl său căutări după cuvinte cheie specifice sitului Alta Vista.

Căutarea după categorii funcționează mai bine atunci când cunoașteți câte ceva despre subiect și puteți estima la ce categorie să căutați informația. În această situație, pur și simplu vă plimbați în josul listei până ce ajungeți la subiectul dorit. Această metodă are marele avantaj că toate temele ce se potrivesc cu acel subiect vor fi găsite în locul unde îl puteți vedea.

Dar ce faceți atunci când nu știți nimic? Nimic în afară de un concept sau chiar numai un cuvânt. Cum veți găsi ceva atunci?

Ca un experiment, încercați să vă gândiți la o problemă, una deja prezentată în această anexă, pentru a ști unde trebuie să ajungeți, cunoscând deja destule lucruri pentru a fi siguri că nu am „trucat" rezultatul. Să zicem că sunteți un programator din Web cu lucrări artistice originale pe situl dumneavoastră și doriți să fiți sigur că nici un individ fără scrupule nu vă poate fura munca pentru a o prezenta ca fiind a lui. Ce veți face? Nu puteți încuia lucrările, trebuie să fie vizibile pe sit, nu-i așa? Nu-i veți pune în față un marcaj uriaș deoarece ar fi urât și, din nou, la ce bun? Decideți să căutați pe Web soluția la această problemă. Folosiți mecanismul de căutare Alta Vista pentru că e cel mai cunoscut.

De unde începeți? Începeți să căutați cuvântul *art* și descoperiți 7.542.680 de potriviri, un lucru nu foarte încurajator. Încercați *art* și *protect*, nu e mai bine, cu 439.330 de documente, sunt prea multe căutări. Dar stați puțin, sunteți interesat de artă digitală, nu artă în general, așa că decideți să adăugați cuvântul *digital*. Bang! 499.580 de potriviri! Mai mult, nu mai puțin! Ce se întâmplă? Așteptați – primele potriviri au câteva cuvinte interesante, „Watermarking and Digital Signature: Protect your work! (Filigranare și semnătură digitală: Protejați-vă munca!)". Se pare că ați găsit ce căutați.

Există o mulțime de moduri pentru a ajusta o căutare; adăugarea unor cuvinte este doar unul dintre ele. Puteți fi mai explicit, programând motorul de căutare să ceară anumite cuvinte sau fraze, prin adăugarea unui semn „plus" în fața cuvântului sau să excludeți cuvinte, prin inserarea unui semn „minus" în fața cuvântului, deși sintaxa este uneori diferită pentru diverse motoare de căutare. Încercați cu `+watermark +digital signature`. Acestea sunt cuvinte găsite în căutarea dumneavoastră, nu unele despre care știați deja. Veți găsi 39 de documente și toate par să fie potrivite pentru ceea ce doreați. Totuși, nu sunteți interesați în mod special de ofertele companiilor comerciale, astfel că îngustați câmpul prin adăugarea unor semne minus pe prima pagină și bingo! Aveți o listă de 28 de documente, doar trei pagini de căutat.

Alta Vista are de asemenea agenți inteligenți care vă permit să realizați căutări complexe în funcție de asocierea cu casete de validare, dar de obicei am observat că vă puteți descurca la fel de bine și obține rezultatele mai repede prin simpla ajustare a ideii inițiale. Există o mulțime de motoare de căutare, fiecare cu facilitățile lui; unele sunt prezentate în secțiunea următoare. Succes la vânătoare!

Motoare de căutare

Motoarele de căutare! Ce mult le îndrăgim, hai să numărăm de câte ori .... Un motor de căutare bun poate atrage utilizatori pe situl dumneavoastră, dacă aveți ceva ce merită văzut și construiți o bună introducere. Trebuie să știți că mulți păianjeni folosesc meta-informațiile pentru a construi o descriere, dar mulți iau doar primele rânduri de text și reușesc să facă ambele treburi bine. Tratați o pagină Web ca pe un articol de ziar cu titluri șocante și un articol de fond pe lângă o descriere formală.

De asemenea, nu uitați că mulți indivizi fără scrupule se angajează în discuții inutile pe Web, împânând paginile lor cu termeni care nu sunt la obiect sau chiar cu duplicate ale termenilor, adesea invizibile, pentru a crește numărul de potriviri la cuvintele căutate și făcând sute de copii ale paginilor lor, astfel utilizatorii care caută un termen obișnuit vor obține o mulțime de pagini cu liste. Aceasta este o practică necinstită, distructivă și periculoasă, iar cei care o fac merită să fie pedepsiți. Puteți evita mulți dintre acești ticăloși în căutările dumneavoastră, prin inserarea unei negări a unei mici liste de cuvinte, de obicei cele vulgare și sexuale, pe care „inteligenții" le includ în paginile lor.

Printre practicanții acestui lucru neserios, veți găsi de la avocați (să nu spuneți că nu e așa!) până la creatori de pagini Web (rușine!) și vânzători de mingi de golf. Nu „cumpărați" nimic de la acești neserioși și poate vor renunța. Bineînțeles că dumneavoastră, stimate cititor, nu veți fi niciodată un pisălog, dar rețineți că o persoană care vă face intenționat să vă pierdeți timpul, nu vă este prieten, indiferent de ce spune. Nu este nevoie să fiți un savant ca să vă dați seama de acest lucru și veți fi surprins când veți afla câți utilizatori manierați urăsc aceste tactici meschine.

Tabelul G.21 rezumă forța motoarelor de căutare prezentate în această secțiune.

Tabelul G.21 Mecanisme de căutare.

Produs	*Tip*	*Audiență*	*Gama de preț*	*Plaforme*
Alta Vista	Cuvânt cheie	Toți utilizatorii	Gratis	Oricare
Excite	Cuvânt cheie	Toți utilizatorii	Gratis	Oricare
Infoseek	Temă/ Cuvânt cheie	Toți utilizatorii	Gratis	Oricare
Lycos	Cuvânt cheie	Toți utilizatorii	Gratis	Oricare
Yahoo!	Temă/Cuvânt cheie	Toți utilizatorii	Gratis	Oricare
Magellan	Cuvânt cheie/ Evaluare	Toți utilizatorii	Gratis	Oricare
Lexis-Nexis	Cuvânt cheie Știri/ Cuvânt cheie	Profesioniști	$$/lună	Oricare
Reuters		Profesioniști	Variază	Oricare

Următoarele secțiuni oferă o scurtă exemplificare a celor mai importante motoare de căutare.

Alta Vista

`http://www.altavista.digital.com/`

Este un sit uriaș de căutare după cuvinte cheie, care are marea problemă a limitării căutării pentru a reduce numărul de potriviri la un număr de documente pe care să le puteți citi într-o viață. Oferă modalități foarte bune de a include opțiuni avansate la multe dintre motoarele de căutare, ca mici adăugări rapide la linia de comandă. Nu vă place să scrieți ghilimelele pentru a marca o frază? Încercați liniuța de subliniere. Adăugați sau eliminați cuvinte folosind simbolurile plus și minus. Foarte logic.

Excite

`http://www.excite.com/`

Motorul de căutare Excite seamănă mult cu Alta Vista, este un sit de căutare după cuvinte cheie, cu o metodă de excludere sau includere a termenilor încorporați. Diferă prin faptul că prezintă o mică selecție de moduri de limitare a căutării pe pagina de rezultate și oferă părerea sa despre relevanța (un termen cam lunecos) paginii referitor la căutarea dumneavoastră. Toate fac acest lucru, încercând să pună „cele mai bune" rezultate cât mai sus, dar Excite vă spune și cât de sigure sunt rezultatele. Excite mai face ceva în plus, are un mecanism inteligent care va lărgi căutarea la termenii înrudiți, așa că dacă veți introduce `elderly people` (persoane în vârstă), se va gândi atât la pensionari, cât și la bătrâni. Dacă acest lucru e bun sau supărător, depinde de cum vă simțiți în ziua respectivă.

Infoseek

`http://www.infoseek.com/`

Infoseek este un motor de căutare atât după cuvinte cheie, cât și după temă și marchează cu mici semne roșii siturile pe care le recomandă, indiferent de ce înseamnă. Deși nu permite negare la „linia de comandă" (deoarece folosește cratima pentru a reprezenta un spațiu fără întrerupere), vă permite să transportați rezultatele de la o căutare la alta, așa că puteți căuta `dog - Dalmatian` pentru a fi siguri că nu veți găsi potriviri cu Dalmatia, care este de fapt actuala Croație.

Lycos

`http://www.lycos.com/`

Lycos este un motor de căutare după cuvinte cheie combinat cu un fel de revistă tematică on-line. Poate fi o resursă foarte bună deoarece puteți găsi adesea un articol care vă spune suficient despre subiectul dumneavoastră pentru a vă oferi cuvintele ajutătoare pe care să le căutați. La fel ca și Alta Vista, puteți include sau exclude cuvinte prin inserarea înaintea lor a semnelor plus sau minus.

Yahoo!

`http://www.yahoo.com/`

Este un sit bazat pe tematică și un foarte bun loc de pornire dacă știți destule despre subiectul căutării dumneavoastră și puteți trece în revistă ușor arborele subiectului. Yahoo! oferă o modalitate de a renunța la categorii prin metodele unei căutări Alta Vista după cuvinte cheie sau puteți parcurge arborele căutării și să găsiți ceea ce se află la capete. Știu că aceasta pare puțin peste mână, dar nimeni nu a spus că programatorii (și eu sunt unul dintre ei) au simț practic. Are un fel de logică inversă (privit din punct de vedere virtual) și este foarte ușor de afișat, așa că aproape de fiecare dată când vedeți un arbore logic, rădăcina va fi în vârf. Totul se trage de la „Alice în Țara Minunilor".

Magellan

`http://www.mckinley.com/`

Acesta este un motor de căutare interesant deoarece este unul dintre puținele care oferă opțiunea de vizionare a unor situri evaluate și recenzate, cât și a așa-numitor situri „de undă verde", care sunt adecvate pentru copii. Cred că majoritatea utilizatorilor înțeleg și sunt de acord cu ceea ce înseamnă aceste cuvinte, dar nu se știe niciodată. Acesta este un sub-motor al sitului Excite, dar nu acceptă încă situri noi pentru a fi analizate. La fel ca și Excite, oferă o estimare a relevanței.

Lexis-Nexis

`http://www.lexis.com/`

Aceasta este o grupă total diferită de motoare de căutare și doar cel mai cunoscut dintre mulții furnizori de articole numai cu text și studii pe Web pentru care se plătește. Acest motor de căutare particular este specializat în căutarea în domeniul juridic, oferind întregul text al cazurilor de proces, legi, aplicațiile cu licență, știri juridice și orice alte informații care ar putea încânta un avocat. Vă înscrieți contra unei taxe lunare pentru fiecare avocat din biroul dumneavoastră, deci este un mod relativ scump de căutare. Ei au câteva servicii gratis, care sunt disponibile pentru public, cum ar fi găsirea avocaților din toată lumea. Avantajul unui abonament constă în faptul că informațiile disponibile pe Lexis-Nexis nu sunt disponbile deloc în altă parte pe Web, deși ați putea vizita secțiunea cu cărți juridice a bibliotecii locale și să găsiți informațiile gratis, plătind doar pentru copierea paginilor care vă sunt necesare la un copiator sau chiar scriindu-le gratis de mână.

Reuters

`http://www.reuters.com/`
`http://www.reutershealth.com/`

Acesta este un alt serviciu cu plată, având unele oferte gratis, cum ar fi știrile de mare interes, ca teme de reflecție. Reuters, după se știe, este cea mai mare agenție de știri din lume, specializată în transmiterea știrilor de toate felurile. Are oferte pentru „vrăjitorii"

financiari, medici, oameni care lucrează în domeniul publicității și marketingului, firme de transport, industria asigurărilor și alte categorii, și să nu uităm produsul cel mai vândut, transmiterea știrilor către ziare, posturi de radio și televiziune și alte piețe media, inclusiv Web.

Puteți accesa Reuters printr-un sit care folosește datele sale sau puteți să vă abonați singuri pentru o mulțime de dolari. Spre deosebire de modelul plata-mai-mult-sau-mai-puțin-destinatar al sitului Lexis-Nexis, Reuters este în principal bazat pe abonament. Unele porțiuni, totuși, cum ar fi situl Reuters Health, permit căutări de informații referitoare la sănătate. Când vă abonați la o sursă de știri Reuters, obțineți tot ceea ce este transmis, iar știrile le puteți folosi sau răspândi, folosind criterii interne. Multe situri mari folosesc Reuters sau altă astfel de agenție de știri pentru a realiza o sursă continuă de știri pe care le publică pentru a atrage oamenii interesați de orice, de la știri sportive și medicale până la cele mai noi bârfe despre ștabii din Silicon Valley.

Există mult mai multe situri, unele cu subiecte bine delimitate, cât și versiuni ale acelor motoare de căutare pe care le găsiți de obicei pe siturile particulare care indexează doar pagini locale.

Index

Simboluri

! atributul textalign (foile de stil în cascadă), 287
! etichete (< OBJECT > atributul codetype), 216
#exec, comanda (aplicațiile CGI), 359
#FIXED, cuvânt cheie, 45-46
#IMPLIED, cuvânt cheie, 46
#PCDATA, cuvânt cheie, 43-44
#REQUIRED, cuvânt cheie, 46
%HTML.Deprecated, 46
& (ampersand), 43-44, 877
&shy, entități caracter mnemonice, 638
&zwj, entități caracter mnemonice, 637
&zwnj, entități caracter mnemonice, 637
* (asterisc) indicator de apariție, 43-44, 876-877
+ (semnul plus) indicator de apariții, 43-44, 876
, (virgulă) conector, 43
- (cratimă), 876-877
/ (operator de împărțire), 876
^ (accent circumflex), 876-877
3D Site, 909
3D Studio Max, 907
<!- - . . . -> etichetă, 684
<!- -comentariu- -> etichetă, 737
< (semnul mai mic decât), 876-877
<= operator, 876-877
<> operator, 876-877

 etichetă, 85-86, 724-725
<elementul TABLE>, 33
= (semnul egal), 871, 876-877
> (semnul mai mare decât), 876-877
>= operator, 876-877
? (semnul de întrebare) indicator de apariții, 43-44

A

accent circumflex (^), 876-877
acronime, 90
active hyperlinks, culori, 502
ActiveX, 347-348
  controale, inserare, 213, 217
  controlul pentru conștientizarea datelor, 443
  liste poștale, 934
  medii avansate, 487
  resurse, 348-350
  VBScript, 347-348
acțiuni (reguli de construcție XSL), 610-611
administratori de sit
  Cold Fusion, 96
  FrontPage, 895
  NetObjects Fusion, 893-894
  SiteMill, 893-894
  tabel de comparare, 893
  WebberActive, 895
Adobe
  Acrobat, 173-174
  Ilustrator, 900
  PageMill, 890
  Photoshop 4.0, 900
    palete de culori personalizate, 502
    filtre, 510
    paletă de culori protejată, 498
    tehnici de iluminare, 510
  SiteMill, 893-894
adresarea legăturilor XML, 575, 834
adrese URL absolute, 156
adrese, formatare, 94
adresele URL de bază, 158-160
afișarea tabelelor, 127-129
agățare (outdentare) (text), 241-242
agent de utilizator, 399
  evenimente
    barbotarea evenimentelor, 403
    rutine de ascultare a evenimentelor, 401
    standarde, 400
    vezi și browsere, 346-347
  legarea datelor, 437-438
Albuquerque Alibi, 492
Alchemy Mindworks GIF Construction Set, 219, 902
alegere
  browsere
    Amaya, 886
    Ariadna, 886
    Cello, 887
    Communicator, 884-885
    Cyberdog, 886
    Emacs W3, 886
    Grail, 886
    HotJava, 886
    Internet Explorer, 883-884
    Lynx, 886
    Mosaic, 884-885
    Navigator, 884-885
    Opera, 885-886
    tabel de comparare, 882-883
    Tango, 886
  editori HTML
    Amaya, 890
    BBEdit, 890
    Composer, 889
    Emacs, 893

Home Page, 890
HomeSite, 890
Hot Dog Express, 889
Hot Dog Pro, 890
HoTMetaL Pro, 889
Multi-Edit, 892
PageMill, 890
tabel de comparare, 888-889
Vi, 893
Visual SlickEdit, 891
Webber, 35, 891
WebEdit Pro, 891
manageri de sit
Cold Fusion, 895
Front page, 895
NetObjects Fusion, 893-894
SiteMill, 893-894
tabel de comparare, 893
WebberActive, 895
servere Web, 917
Apache, 919-920
CERN HTTPD, 918
CL-HTTP, 919-920
Cold Fusion, 921
Jigsaw, 919-920
Lotus Notes/Domino, 920
MacHTTPD, 918-919
NSCA HTTPT, 918-919
O'Reilly WebSite 2.0, 918-919
SuiteSpot, 919-920
Sun server family, 919-920
tabel de comparare, 918
WebStar, 918-919
unelte de mapare a sitului, 895
CLEARWeb, 896
Exploder Site Map, 897
Governor Site Map, 897
HotSauce, 897
Joe Bart's Site Map, 897
MAPA Service, 896
MapXsite, 896
SiteAnalyst, 896
SiteCommand, 897
tabel de comparare, 895
Visual Site Map, 897
WebAnalyzer, 897
WebCollection, 897
WebCutter, 897
validatori HTML, 915-916
Alfabet: istorie, evoluție și proiectarea literelor pe care le folosim astăzi, 538
alfabetul chirilic (ISO 8859-5), 625
alfabetul ebraic (ISO 8859-8), 625
alfabetul grecesc (ISO 8859-7), 625
aliniere
cadre, 282-283
celule (tabele), 124
coloane, 16
imagini, 50-52, 148-149, 514
imagini din fundal, 821
aliniere pe orizontală, 523
aliniere pe verticală, 522
aliniere una lângă alta, 523
imagini încorporate, 525-526
internaționalizarea limbajului HTML, 638
paragrafe, 85, 522
tabele, 116
denumire, 124
rânduri, 118
text
aliniere la dreapta, 520
aliniere la stânga, 520
aliniere pe centru, 521-523
foi de stil în cascadă (CSS), 241-242
aliniere la dreapta
imagini, 523
paragrafe, 522
text, 521
aliniere la stânga
imagini, 523
paragrafe, 522
text, 521
alinierea imaginilor pe verticală, 522
alinierea orizontală a imaginilor, 523
Allaire's HomeSite 2.5, 890
American Cybernetics Multi-Edit, 892
AMSD Ariadna, 886
analizoare (SGML), 38-39
ancora destinație (hiperlegături), 168
ancora sursă (hiperlegături), 168
ancore (hiperlegături), 169-170
ancora destinație, 168
ancora sursă, 168
ancore denumite, 170-171
animație
DHTML, scriptul de tratare a evenimentului, 418, 422
fișiere GIF
creare, 219
inserare, 219-220
inserare, 213-215
layere, 299
navigarea pe pagini Web, 486-487
ANSI (Institutul American de Standarde), standarde pentru limbajul de marcare, 36
antet (document HTML), 31-32
anteturi
anteturi HTTP, 354
câmpuri de cerere, 354-355
câmpuri de răspuns, 355-356
celule (tabele), 118-120
considerații de proiectare, 471
documente HTML, 70-71
niveluri, 82
stiluri, 82
anulare
foi de stil XSL, 616-617
text bidirecțional, 637
apăsare
evenimentul onclick, 405

evenimentul ondblclick, 405
apelarea aplicațiilor CGI, 358
eticheta <A>, 358
eticheta <FORM>, 358
eticheta <IMG>, 358
scripturi de partea clientului, 359
server-side include, 359
aplicații
CGI
apelare, 358-359
aplicații pe partea serverului, 359-360
cereri HTTP, 356
integrarea bazelor de date, 370-375
integrarea în situl Web, 354
motoare de căutare, 360-361
numărătoare de acces, 361-363
returnarea datelor din aplicații CGI, 358
rutine de tratare a formularelor, 358, 363, 368-370
trimiterea datelor la aplicații CGI, 356-358
conștientizarea datelor, 435
inserare, 216
XML
Cartea de răspunsuri, 580
Channel Definition Format(CDF), 580
Limbajul de marcare chimic, 580
Limbajul de marcare cu dispozitiv (HDML), 580
Limbajul de marcare matematic (MathML), 580
Open Software Description (OSD), 580
OpenTag, 580
AppleScript, 24, 353
Archie, 24
Ariadna, 886
aritmetic
funcții, 870
operatori, 867
armonia culorilor: ghid de combinare creativă a culorilor, 502
ASCII pe 7-biți, internaționalizarea limbajului HTML, 624-625
ASCII US și internaționalizarea limbajului HTML, 624-625
ascundere
informații de la motoare de căutare, 668
layere, 305-307
Asociația americană a editurilor, 36
Asociația Europeană pentru Standardizarea Informației și Sisteme de Comunicații, 579, 617-618
Asociația Europeană pentru Standardizarea Informației și Sisteme de Comunicații (ECMA), 392
asociere
etichete și controale din formulare
asociere explicită, 208-209
asociere implicită, 208-209
legături XML cu informația, 574-575, 833
asterisc (*) indicator de apariții, 43-44, 876-877
Asymmetrix Multimedia Toolbook, 907
Atkinson, Bill, 21
atribuirea de clase etichetelor HTML, 248-249
atribute
atribute de aliniere, 68
atribute internaționale, 68, 733
atribute nerecomandate, 46, 671
atributele core, 68, 732-733
atributul accept-charset (eticheta <FORM>), 196
atributul accesskey, 657
eticheta < AREA>, 184
eticheta < INPUT>, 211-212
atributul action (eticheta <FORM>), 195, 358
atributul ACTUATE (XML), 575, 833-834
atributul align
<TABLE>, 116
eticheta <FIELDSET>, 209-210
eticheta <IFRAME>, 282-283
eticheta <IMG>, 148, 514, 522
eticheta <TD>, 124
eticheta <TR>, 118
problema accesibilității, 650-651
atributul alt
eticheta <AREA>, 184, 654
eticheta <IMG>, 147-148, 215, 513, 652
eticheta <INPUT>, 203-204
interpretarea browserului, 763
atributul axes (eticheta <TD>), 123
atributul background, 821
atributul background-color, 820
atributul background-image, 820
atributul background-position, 821
atributul background-repeat, 821
atributul bgcolor (eticheta <BODY>), 77, 502
atributul border, 825
atributul border-bottom, 823

atributul border-bottom-width, 824
atributul border-color, 824
atributul border-left, 823
atributul border-left-width, 824
atributul border-right, 823
atributul border-right-width, 824
atributul border-style, 824
atributul border-top, 823
atributul border-top-width, 824
atributul border-width, 824
eticheta <IMG>, 151, 514
eticheta <TABLE>, 117
atributul cellpadding (eticheta <TABLE>), 117
atributul cellspacing (eticheta <TABLE>, 117
atributul center
eticheta <DIV>, 520
eticheta <P>, 522
atributul class, 68
atributul classid (eticheta <OBJECT>), 216-217, 438
atributul classname (eticheta <OBJECT>), 216
atributul clear (eticheta <BR>), 85-86, 825
atributul codebase (eticheta <OBJECT>), 216-217
atributul codetype (eticheta <OBJECT>), 216-218
atributul color (eticheta <FONT>), 536, 820
atributul cols
eticheta <TABLE>, 116
eticheta <TEXTAREA>, 208
atributul colspan (eticheta <TD>), 120
atributul compact (<OL> și <UL>), 111
atributul CONTENT-ROLE (XML), 574
atributul CONTENT-TITLE (XML), 574
atributul controls (eticheta <IMG>), 220
atributul data (eticheta <OBJECT>), 214
atributul DATAFLD, 456
atributul DATAFORMATAS, 456-457
atributul DATAPAGESIZE, 457
atributul dir, 68, 637
atributul disabled (eticheta <INPUT>), 203-204
atributul display, 826
atributul dynsrc (eticheta <IMG>), 220
atributul face (eticheta <FONT>), 534
atributul float, 825
atributul font, 820
atributul font-family, 238-240, 818
atributul font-size, 239-240, 819
atributul font-style, 240, 819
atributul font-variant, 819
atributul font-weight, 240, 819
atributul for (eticheta <LABEL>), 208-209
atributul frame (eticheta <TABLE>), 116
atributul frameborder (eticheta<FRAME>), 277
atributul height, 825
atributul HREF (XML), 575, 833
atributul href
eticheta <A>, 169-170
eticheta <BASE>, 73
eticheta <LINK>, 74
legături XML, 573
atributul hspace (eticheta <IMG>), 151, 514, 525
atributul http-equiv
atributul name, 774
eticheta <META>, 75-77
atributul id, 15, 68
eticheta <A>, 170
eticheta <OBJECT>, 438
atributul img (eticheta <IMG>), 513
atributul INLINE (XML), 574-575, 833
atributul ismap (eticheta <IMG>), 152-153, 185
atributul lang, 68, 635
atributul language (eticheta <SCRIPT>), 317-318, 335
atributul left
eticheta <DIV>, 520
eticheta <IMG>, 523
eticheta <LAYER>, 301
eticheta <P>, 522
atributul leftmargin (eticheta <BODY>), 519
atributul letter-spacing, 821
atributul line-height, 241, 822
atributul link (eticheta <BODY>), 502
atributul list-style, 826
atributul list-style-image, 826
atributul list-style-position, 826
atributul list-style-type, 826
atributul loop (eticheta <IMG>), 220
atributul margin, 823
atributul margin-left, 241-242
atributul margin-right, 241-242
atributul margin-top, 241-242
atributul marginheight (eticheta <FRAME>), 277
atributul marginwidth (eticheta <FRAME>, 277

atributul media (eticheta <LINK>), 74
atributul method (eticheta <FORM>), 195
atributul multiple (eticheta <SELECT>), 206-207
atributul name
atributul http-equiv, 774
eticheta <A>, 170
eticheta <FRAME>, 276-277
eticheta <MAP>, 181
eticheta <META>, 75-77
atributul nohref (eticheta <AREA>), 184
atributul noresize (eticheta <FRAME>), 276-277
atributul nowrap (eticheta <TD>), 124
atributul number (eticheta <FONT>), 226-228
atributul onreset (eticheta <FORM>), 196
atributul onsubmit (eticheta <FORM>), 196
atributul padding, 823
atributul padding-bottom, 823
atributul padding-left, 823
atributul padding-right, 823
atributul padding-top, 823
atributul paramName (eticheta <PARAM>), 216
atributul paramValue eticheta (<PARAM>), 216
atributul prompt (eticheta <ISINDEX>), 205
atributul readonly (eticheta <TEXTAREA>), 208-209
atributul rel
eticheta <A>, 169-170
eticheta <LINK>, 74
atributul rev
eticheta <A>, 169-170
eticheta <LINK>, 74
atributul right
eticheta < IMG>, 523
eticheta <DIV>, 520
eticheta <P>, 522
atributul ROLE (XML), 574-575, 833
atributul rows (eticheta <TEXTAREA>), 208
atributul rowspan (eticheta <TD>), 120
atributul rules (eticheta <TABLE>), 117
atributul scrolling (eticheta <FRAME>), 276-277
atributul selected (eticheta <OPTION>), 206
atributul shape (eticheta <AREA>), 181-183
atributul shapes (eticheta <OBJECT>), 184-185
atributul SHOW (XML), 575, 833-834
atributul size
eticheta <FONT>, 226-228, 535
eticheta <SELECT>, 206
atributul src
eticheta <BGSOUND>, 220
eticheta <EMBED>, 219
eticheta <FRAME>, 276-277
eticheta <IMG>, 147-148, 513
atributul start
eticheta <IMG>, 220
eticheta <OL>, 104-108
atributul style, 68
atributul tabindex, 656-657
eticheta <AREA>, 184
eticheta <INPUT>, 211
atributul target
eticheta <A>, 169-170
eticheta <AREA>, 184
eticheta <BASE>, 280
eticheta <FRAME>, 278-280
atributul text (eticheta <BODY>), 502-503
atributul text-align, 241-242, 822
atributul text-decoration, 241, 821
atributul text-indent, 241-242, 822
atributul text-transform, 822
atributul textalign
foile de stil în cascadă (CSS), 287
foile de stil JavaScript, 287
atributul title, 68
atributul TITLE (XML), 574-575, 833
atributul top (eticheta <LAYER>), 301
atributul topmargin (eticheta <BODY>), 519
atributul type
eticheta <INPUT>, 196-197
eticheta <LINK>, 74
eticheta <OBJECT>, 214
eticheta <OL>, 106-108
eticheta <PARAM>, 216
eticheta <UL>, 109
atributul usemap
eticheta <IMG>, 180
eticheta <INPUT>, 205
eticheta <MAP>, 153
atributul valign (eticheta <TR>), 118
atributul valuetype (eticheta <PARAM>), 216
atributul visibility (eticheta <LAYER>), 305
atributul vspace (eticheta <IMG>), 151, 514, 525
atributul white-space, 124, 826
atributul width, 825
eticheta <IFRAME>, 282-283

eticheta <IMG>, 150-151, 478, 514
eticheta <PRE>, 87
eticheta <TABLE>133
atributul word-spacing, 821
atributul XML-LINK (XML), 573, 578, 832-833
atributul XML-SPACE (XML), 569-570, 828
attributul background-attachment, 821
cele care manevrează evenimentele, 68
declarații în limbajul XML, 829
definiția, 29
elemente, 45-47
eticheta <IFRAME>, 282-283
eticheta <IMG>, 150-151, 478, 514
liste, 100, 108, 111
prestabilite în limbajul XML, 830
selectori, 816-817
tabelul de comparare a browserelor, 736-880
tipurile enumerate (XML), 569-570, 830
tipurile șir (XML), 569, 830
tipurile token (tipurile simbolice) (XML), 569-570, 830
vertical align attribute, 822
XML DTDs, 569
atributul pentru spațiu alb, 569-570
predefinite, 569-570
remapare, 569
versus elements, 593-595
atribute de aliniere, 68
atribute internaționale, 68
atribute pentru clasificare
atributul display, 826
atributul list-style, 826
atributul list-style-image, 826
atributul list-style-position, 826
atributul list-style-type, 826
atributul white-space, 826
atribute rel
eticheta <A>, 169-170
eticheta <LINK>, 74
atributele core, 45, 68, 732-733
atributul accept-charset (eticheta <FORM>), 196
atributul accesskey, 657
eticheta <AREA>, 208-209
eticheta <INPUT>, 211-212
atributul action (eticheta <FORM >), 195, 358
atributul ACTUATE (XML), 575, 833-834
atributul align
<TABLE>, 116
eticheta <FIELDSET>, 209-210
eticheta <IFRAME>, 283
eticheta <IMG>, 148, 514, 522
eticheta <TD>, 124
eticheta <TR>, 118
problema accesibilității, 650-651
atributul alt
eticheta <AREA>, 184, 654
eticheta <IMG>, 147-148, 215, 513, 652
eticheta <INPUT>, 203-204
interpretarea browserelor, 763
atributul axes (eticheta <TD>), 123
atributul background, 821
atributul background-attachment, 821
atributul background-color, 820
atributul background-image, 820
atributul background-position, 821
atributul background-repeat, 821
atributul bgcolor (eticheta <BODY>), 77, 502
atributul border, 825
atributul border-bottom, 823
atributul border-bottom-width, 824
atributul border-color, 824
atributul border-left, 823
atributul border-left-width, 824
atributul border-right, 823
atributul border-right-width, 824
atributul border-style, 824
atributul border-top, 823
atributul border-top-width, 824
atributul border-width, 824
eticheta <IMG>, 151, 514
eticheta <TABLE>, 117
atributul cellpadding (eticheta <TABLE>), 117
atributul center
eticheta <DIV>, 520
eticheta <P>, 522
atributul class, 68
atributul classid (eticheta <OBJECT>), 216-217, 438
atributul classname (eticheta <OBJECT>), 216
atributul clear (eticheta <BR>), 85-86, 825
atributul codebase eticheta(<OBJECT>), 216-217
atributul codetype, (funcția <OBJECT>), 216-218
atributul color, 536, 820
atributul cols
eticheta <TABLE>, 116
eticheta <TEXTAREA>, 208
atributul colspan (eticheta <TD>), 120
atributul compact (<OL> și <UL>), 111

atributul CONTENT-ROLE (XML), 574
atributul CONTENT-TITLE (XML), 574
atributul controls (eticheta <IMG>), 220
atributul data (eticheta <OBJECT>), 214
atributul DATAFLD, 456
atributul DATAFORMATAS, 456-457
atributul DATAPAGESIZE, 457
atributul dir, 68, 637
atributul disabled (eticheta <INPUT>), 203-204
atributul display, 826
atributul dynsrc (eticheta <IMG>), 220
atributul enctype
eticheta <FORM>, 195-196
suport browser, 757
atributul face (eticheta <FONT>), 534
atributul float, 825
atributul font, 820
atributul font-family, 238-240, 818
atributul font-size, 239-240, 819
atributul font-style, 240, 819
atributul font-variant, 819
atributul font-weight, 240, 819
atributul for (eticheta <LABEL>), 208-209
atributul frame (eticheta <TABLE>), 116
atributul frameborder (eticheta <FRAME>), 277
atributul height, 825
eticheta <IFRAME>, 282-283
eticheta <IMG>, 150-151, 478, 514
atributul href
eticheta <A>, 169-170
eticheta <BASE>, 73
eticheta <LINK>, 74
XML, 574, 833
atributul hspace (eticheta <IMG>), 151, 514, 525
atributul http-equiv
atributul name, 774
eticheta <META>, 75-77
atributul id, 15, 68
eticheta <A>, 170
eticheta <OBJECT>, 438
atributul img (eticheta <IMG>), 513
atributul INLINE (XML), 574-575, 833
atributul ismap (eticheta <IMG>), 152-153, 185
atributul lang, 68, 635
atributul language (eticheta <SCRIPT>), 317-318, 335
atributul left
eticheta <DIV>, 520
eticheta <IMG>, 523
eticheta <LAYER>, 301
eticheta <P>, 522
atributul left margin (eticheta <BODY>), 519
atributul letter-spacing, 821
atributul line-height, 241, 822
atributul link (eticheta <BODY>), 502
atributul list-style, 826
atributul list-style-image, 826
atributul list-style-position, 826
atributul list-style-type, 826
atributul loop (eticheta <IMG>), 220
atributul margin, 823
atributul margin-left, 241-242
atributul margin-right, 241-242
atributul margin-top, 241-242
atributul marginheight (eticheta <FRAME>), 277
atributul marginwidth (eticheta <FRAME>), 277
atributul media (eticheta <LINK>), 74
atributul metod (eticheta <FORM>, 195
atributul multiple (eticheta <SELECT>), 206-207
atributul name
atributul http-equiv, 774
eticheta <A>, 170
eticheta <FRAME>, 276-277
eticheta <MAP>, 181
eticheta <META>, 75-77
atributul nohref (eticheta <AREA>), 184
atributul noresize (eticheta <FRAME>), 276-277
atributul nowrap (eticheta <TD>), 124
atributul number (eticheta <FONT>), 226-228
atributul onreset (eticheta <FORM>), 196
atributul onsubmit (eticheta <FORM>), 196
atributul padding, 823
atributul padding-bottom, 823
atributul padding-left, 823
atributul padding-right, 823
atributul padding-top, 823
atributul paramName (eticheta <PARAM>), 216
atributul paramValue (eticheta <PARAM>), 216
atributul prompt (eticheta <ISINDEX>), 205
atributul readonly (eticheta <TEXTAREA>), 208-209
atributul rev
eticheta <A>, 169-170
eticheta <LINK>, 74
atributul right
eticheta <DIV>, 520
eticheta <IMG>, 523
eticheta <P>, 522
atributul ROLE (XML), 574-575, 833
atributul rows (eticheta <TEXTAREA>), 208
atributul rowspan (eticheta <TD>), 120

atributul rules (eticheta <TEXTAREA>), 117
atributul scrolling (eticheta <FRAME>), 276-277
atributul selected (eticheta <OPTION>, 206
atributul shape (eticheta <AREA>), 181-183
atributul shape (eticheta <OBJECT>), 184-185
atributul SHOW (XML), 575, 833-834
atributul size
eticheta <FONT>, 226-228, 535
eticheta <SELECT>, 206
atributul src
eticheta <BGSOUND>, 220
eticheta <EMBED>, 219
eticheta <FRAME>, 276-277
eticheta <IMG>, 147-148, 513
atributul start
eticheta <IMG>, 220
eticheta <OL>, 104-108
atributul style, 68
atributul tabindex, 656-657
eticheta <AREA>, 184
eticheta <INPUT>, 211
atributul target
eticheta <A>, 169-170
eticheta <AREA>, 184
eticheta <BASE>, 280
eticheta <FRAME>, 278-280
atributul text (eticheta <BODY>), 502-503
atributul text-align, 241-242, 287, 822
atributul text-decoration, 241, 821
atributul text-indent, 241-242, 822
atributul text-transform, 822
atributul title, 68
atributul TITLE (XML), 574-575, 833
atributul top (eticheta <LAYER>), 301
atributul topmargin (eticheta <BODY>), 519
atributul type
eticheta <INPUT>, 196-197
eticheta <LINK>, 74
eticheta <OBJECT>, 214
eticheta <OL>, 106-108
eticheta <PARAM>, 216
eticheta <UL>, 109
atributul usemap
eticheta <IMG>, 180
eticheta <INPUT>, 205
eticheta <MAP>, 153
atributul valign
eticheta <TD>, 124
eticheta <TR>, 118
atributul valuetype (eticheta <PARAM>), 216
atributul vertical-align, 822
atributul visibility (eticheta <LAYER>), 305
atributul vspace (eticheta <IMG>), 151, 514, 525
atributul white-space, 124, 826
atributul width, 825
<TABLE>, 116
eticheta <IFRAME>, 282-283
eticheta <IMG>, 150-151, 478, 514
eticheta <PRE>, 87
atributul word-spacing, 821
atributul XML-ATTRIBUTES (XML), 569, 832-833
atributul XML-LINK (XML), 573, 578, 832-833
atributul XML-SPACE (XML), 569-570, 828
audiența paginilor Web, 249-250
design, 462-467
forumul DisAbilities, 546-547
motoare de căutare, rol, 660
Auto F/X, 511
auto-încărcare (browsere), 148
autor de pagini Web, identificare, 77
aviziere electronice pentru forumul DisAbilities, 547-548

B

barbotarea evenimentului, (agitarea evenimentului), 397, 403, 735
bare de defilare
bare orizontale de derulare, 481
cadre, 276-277
bare de navigație (cadre), 284-285
Barry's Clip Art Server, 905
baze de date
CGI, 370-375
interogări, 436
motoare de căutare, 662
BBEdit, 890
Berners-Lee, Tim, 22-25, 27-28, 937
Bobby 2.0, 917
Border Manager, 927
borduri
atribute
atributul border, 825
atributul border-bottom, 823
atributul border-bottom-width, 824
atributul border-color, 824
atributul border-left, 823
atributul border-left-width, 824
atributul border-right, 823
atributul border-right-width, 824
atributul border-style, 824
atributul border-top, 823
atributul border-top-width, 824
atributul border-width, 824
atributul clear, 825
atributul float, 825

atributul height, 825
atributul width, 825
cadre, 277
imagini, 151, 514
tabele, 117
broserul vocal pwWebSpeak, 643-644
broșuri (șabloane de pagini stil în cascadă), 257-260
browsere, 882
agent de utilizator, 399
anteturi, probleme cu, 82
barbotarea evenimentului, (agitarea evenimentului), 403, 735
blocări cu eticheta <OBJECT>, 214
browser care nu suportă cadrele și elementul FRAMESET, 32
browsere non-grafice, 148, 184, 394
browsere vocale, 643-644
browserul Amaya, 15, 886
browserul Ariandna, 886
browserul Cello, 887
browserul Lynx, 886
browserul Mosaic, 670, 884-885
browserul Opera, 670, 885-886
browserul Tango, 670, 886
bwoserul Navigator, 671, 884-885
cadru, suport, 273, 282
caracteristica de auto-încărcare, 148
cele mai noi surse de informare, 800
Communicator, 884-885
conștientizarea datelor, 435
coordonatele ferestrelor browserelor, 261
culoare
culorile browserului, 504
paletă de culori protejată, 499-501
Cyberdog, 886
definirea, 29
DHTML
degradare elegantă, 386-387
suport pentru scripturi de tratare a evenimentelor, 432-433
Emacs W3, 886
extensiile fișierelor, 213-214
foile de stil în cascadă (CSS), 230
formulare, alternativă pentru, 203-204
furnizarea fonturilor, 639
Grail, 886
HotJava, 886
HTML+, 25
incompatibilități, 735-736
Internet cu un browser Accent, 670
Internet Explorer, 671, 883-884
agitarea evenimentului, 735
Document Object model(DOM), 735
URL-urile mailto, 163
interpretarea fișierelor, 213-214
istoric, 20
JavaScript
actualizarea proprietăților browserului, 320
detectarea caracteristicilor browserului, 320
suport, 318
layere, 299, 309
legarea datelor, 437-438
margini, îmbinarea cu partea de sus a browserului, 488
pagini de stil (foi de stil), 222, 523
probleme de afișare a textului, 83, 222
seturile de caractere, probleme cu, 633-634
specific pentru o anumită platformă, 671
suport ActiveX, 217
suport Java, 216
suportul fișierelor GIF animate, 219
tabel de comparare, 883-884
tabelul de comparare a browserelor, 736-880
caractere, 802-815
culori, 801
entități escaped, 800
evenimente intrinseci, 799
importanță, 734
precizie, 815
proiectare, 735-736
VBScript
ascunderea scripturilor de browserele incompatibile, 335-336
rularea scripturilor, 338-339
suport, 332-335
browsere care nu suportă cadrele (eticheta <FRAMESET>), 32
browsere nongrafice, 148, 394
browsere vocale, 643-644
browsere Web, vezi browsere
browserul Alis Technology Tango, 886
browserul Amaya, 15, 886, 890
browserul Cello, 887
browserul Communicator, 884-885
browserul Grail, 886
browserul HotJava, 216, 886
browserul Jigsaw Web server, 216, 919-920
browserul Lynx, 394, 886
cele mai noi informații, 800
hărți de imagine, 771
hărți de imagini pe partea clientului, 741
browserul Microsoft Internet Explorer, 883-884
cele mai noi informații, 800

fișiere de sunet, inserare, 220
JavaScript, 288, 318
JScript, 318
suport ActiveX, 217
suport Java, 216
suport pentru fișiere .AVI, 220
browserul Mosaic, 671, 884-885
cele mai noi informații, 800
etichetele <LINK>, 74
browserul Opera, 671, 800, 885-886
browserul Tango, 671, 886
buletine (șabloane CSS), 253-257
Bunge, Robert, 322
Bush, Vannevar, 20, 28
butoane (formulare)
butoane image, 198
butoane push, 197, 205
butoane radio, 200-201
butoane reset, 198, 205
butoane submit, 198, 205
butoane image (formulare), 198
butoane radio (formulare), 200-201
butoane reset (formulare), 205
butoanele submit (formulare), 198, 205

C

C/C++
CGI (Common Gateway Interface), 353, 375-377
resurse online, 924
C++ Lite FAQ, 924
Lysator C Programming, 925
Programmer's Source, 924
CINet Download.com, 898
cadre, 16, 31, 273
bare de defilare, 276-277
cadre inline, 15, 282-284
coloane, 275
comunicarea între cadre, 278-281
conținut alternativ, 281-282
creare, 274
defilare, 276-277
dimensionare, 276-277
elemente de accesibilitate, 657, 758
hărți imagine, 184
hiperlegături, 169-170
index, 286
înălțime, 277
lățimi, 277
legare, 275, 280, 285
machetare, 284, 529
margini, 277
motoare de căutare, 668
navigare
bare de navigare, 284-285
folosirea marginii paginii pentru navigare, 491
folosirea părții de jos a ecranului pentru navigare, 488
folosirea părții superioare a ecranului pentru navigare, 488
programe păianjen, 661
rânduri, 275
relațiile dintre seturile de cadre, 280
stuctură, 32
suportul browserului, 274
tabele, 117
tabla de materii, 285-286
trecerea în revistă a capitolelor, 285-286
video, 220
cadre inline, 15, 282-284
camere de discuții pentru forumul DisAbilities, 547-548
candidați (legături XML), 576-577
capete de paragraf agățate (text), 241-242
capitalizare, 819, 822
caractere
referințe XML, 831
tabelul de comparare a browserelor, 802-815
URL-uri (caractere rezervate sau nesigure), 165
caractere ASCII, 802-808
caractere de control (ISO 646), 624-625
caractere de delimitare, 30
caractere non-Latine (HTML), 37-38
caractere numerice, 802-815
caracteristici nerecomandate, 671
DTDs, 46
etichete, 16-17
caracteristicile obiectului de curs dimensiune-font, 604
caracteristicile obiectului de curs nume-familie-fonturi, 604
caracteristicile obiectului de curs spațiere, 604
caracteristicile obiectului de curs spațiu dintre linii, 604
caracteristicile obiectului de curs spațiu-înainte, 604
care nu face distincție între majuscule și litere mici
atributul id, 170
atributul name, 170
JavaScript, 287, 427
VBScript, 427
Cartea de răspunsuri, 580
casete cu liste (formulare), 205-207
casete de dialog (formulare), 198-199
CAST (Center for Applied Special Technology), 933
CAST Bobby 2.0, 917
căi, 161-162
căutare dupa frază (motoare de căutare), 663
căutare dupa mediu (motoare de căutare), 663
căutări de proximitate (motoare de căutare), 663
căutări prelucrate (motoare de căutare), 663
câmp URI (anteturi HTTP), 356
câmpul ACCEPT (anteturi HTTP), 354
câmpul ACCEPT-ENCODING (anteturi HTTP), 354
câmpul ACCEPT-LANGUAGE (anteturi HTTP), 354
câmpul ALLOWED (anteturi HTTP), 355
câmpul antetului de răspuns (anteturi HTTP), 354-355
câmpul antetului de răspuns (anteturi HTTP), 355-356
câmpul AUTHORIZATION (anteturi HTTP), 355
câmpul CHARGE-TO (anteturi HTTP), 355
câmpul CONTENT-ENCODING (anteturi HTTP), 355
câmpul CONTENT-LANGUAGE (anteturi HTTP 435
câmpul CONTENT-LENGTH (anteturi HTTP), 355
câmpul CONTENT-TRANSFER-ENCODING (anteturi HTTP), 355
câmpul CONTENT-TYPE (anteturi HTTP), 355
câmpul COST (anteturi HTTP), 355
câmpul DATE (anteturi HTTP), 355
câmpul DERIVED-FROM (anteturi HTTP), 355
câmpul EXPIRES (anteturi HTTP), 356
câmpul FROM (anteturi HTTP), 355
câmpul IF-MODIFIED-SINCE (anteturi HTTP), 355
câmpul LAST-MODIFIED (anteturi HTTP), 355
câmpul LINK (anteturi HTTP), 355
câmpul MESSAGE-ID (anteturi HTTP), 356
câmpul PRAGMA (anteturi HTTP), 355
câmpul PUBLIC (anteturi HTTP), 356
câmpul REFERER (anteturi HTTP), 355
câmpul TITLE (anteturi HTTP), 356
câmpul VERSION (anteturi HTTP), 356
câmpuri
anteturi HTTP
ACCEPT, 354
ACCEPT-ENCODING, 354
ACCEPT-LANGUAGE, 354
ALLOWED, 355
AUTHORIZATION, 355
CHARGE-TO, 355
CONTENT-ENCODING, 355
CONTENT-LANGUAGE, 355
CONTENT-LENGHT, 355
CONTENT-TRANSFER-ENCODING, 355
CONTENT-TYPE, 355
COST, 355
DATE, 355
DERIVED-FROM, 355
EXPIRES, 355
FROM, 355
IF-MODIFIED-SINCE, 355
LAST-MODIFIED, 355
LINK, 355
MESSAGE-ID, 356
PRAGMA, 355
PUBLIC, 356
REFERER, 355
TITLE, 356
URI, 356
USER-AGENT, 355
VERSION, 356
formulare de date, creare, 439
celule (tabele), 116
aliniere, 124
atributul cellspacing (eticheta <TABLE>), 117
creare, 118-120
definire, 133
formatare, 116
fuzionarea, 120-122
împachetarea textului, 124
lățimi, 127-128
mărime, setare, 120
padding, 117
situl Census Bureau Web, 444
spațiu, 117
Center for Applied Special Technology (CAST), 933
centrare
paragrafe, 522
text, 520-527
cercuri (design considerations), 531
CERN (European Laboratory for Particle Physics), 22, 36, 185-186
CERN HTTPD, 918
CGI (Common Gateway Interface), 352
alternative de limbaj de programare, 353
aplicații
apelare, 358-359
aplicații pe partea serverului, 359-360
cereri HTTP pentru aplicații CGI, 356
integrarea bazelor de date, 370-375
integrarea în situl Web, 354
motoare de căutare, 360-361

numărătoare de acces, 361-363
returnarea datelor, 358
rutine de tratare a formularelor, 358, 363, 368-370
trimiterea datelor, 356-358
directive de server Web, 358
formulare, 24-25, 357
liste poștale, 934
metode
metoda GET, 357
metoda POST, 357
PERL CGI, 162
resurse
cărți, 377
grupuri de discuții, 377
liste poștale, 377
situri Web, 371-373, 46, 921-927
resurse online, 921
legături referitoare la C/C++, 925
legături referitoare la Perl, 923-924
legături referitoare la Visual Basic, 925-926
Matt's script Archive, 923
Selena Sol's Script Archive, 923
situl CGI Collection, 922
situl CGI Made Really Easy, 922
situl CGI Resource Index, 922
situl CGI Resources, 922
situl NSCA, 922
searchpart, 161-162
variabile de mediu, 356
Channel Definition Format (CDF), 580
Check Point Firewall, 928
Chijiiwa, Hideaki, 502
Cisco Systems PIX, 928
citate, 90
citate inline, 15
citire (SGML), 38-39
cititoare (Acrobat reader), 173-174
cititoare de ecrane, 643-644
CL-HTTP (Common Lisp Hypermedia Server), 919-920
Claris Home Page 3.0, 890
claritatea, principiu de proiectare, 461, 472, 475, 541
clasa LINK (SGML), 38-39
clasa MINIMIZE (SGML), 38-39
clasa obiectului de curs caracter, 602
clasa obiectului de curs casetă, 603
clasa obiectului de curs celulă-tabel, 603
clasa obiectului de curs chenar-tabel, 603
clasa obiectului de curs coloană-tabel, 603
clasa obiectului de curs câmp-linie, 602
clasa obiectului de curs defilare, 602
clasa obiectului de curs grafică externă, 602
clasa obiectului de curs grup-afișare, 604
clasa obiectului de curs întrerupere paragraf, 602
clasa obiectului de curs legătură, 604
clasa obiectului de curs linie despărțitoare orizontală, 602
clasa obiectului de curs linie despărțitoare verticală, 602
clasa obiectului de curs marcare, 602
clasa obiectului de curs paragraf, 602
clasa obiectului de curs porțiune-tabel, 603
clasa obiectului de curs rând-tabel, 603
clasa obiectului de curs secvență, 604
clasa obiectului de curs secvență-pagină-simplă, 604
clasa obiectului de curs tabel, 603
clasa obiectului de curs text înglobat, 602
clasa OTHER (SGML), 38-39
clase
etichete, atribuire, 248-249
foile de stil JavaScript, 288
selectori, separare, 816
SGML
clasa LINK, 38-39
clasa MINIMIZE, 38-39
clasa OTHER, 38-39
clasificarea documentelor după motoarele de căutare, 75
CLEARWeb, 896
client pull, 76-77
clip art (situl Caboodles), 471
codarea caracterelor
entități XML, 832
internaționalizarea limbajului HTML, 624
codificări pe 16 biți, 626-627
codificări pe 8 biți, 625
codul ASCII pe 7 biți, 624-625
combinații de biți, 624-625
ISO 10646, 627-628
setul de codificare a caracterelor, 624
MIME, 628
codări pe 8-biți, internaționalizarea limbajului HTML, 625
codări pe 16-biți, internaționalizare HTML, 626-627
formatarea textului, 91-93
codificarea formularelor, 195-196
JavaScript, inserare, 327-330

coduri de evadare (URL-uri), 165
VBScript, inserare, 337-338
Coffee Cup Image Mapper++, 194
Cold fusion
3.0, 895
server, 920
coloane
cadre, 275
coloanele ziarului, 141-142
conținut, inserare, 138-139
creare, 133-134
grupare, 16
index, coloane, 136-138
lățimi, 134
machetă, 52-56
note, 138-139
spațiu, 134
tabele, 116
alipire peste, 120-121
grupare, 125-127
texte suplimentare, 138-139
coloane stil ziar
creare, 141-142
inserare text, 142-144
comentarii
"no frames", 274
DHTML, scripturi de tratare a evenimentelor, 433-434
JavaScript, 293, 318
SGML, 37-38
XML, 566, 827
Common Gateway Interface, *vezi* CGI
Composer (editor), 889
comprimare (comprimare cu pierderi), 513
comprimare cu pierderi, 513
comunicare
cadre, 278-281
Java și JavaScript, 322-323
JavaScript și VRML, 322-323
comutarea codului (codificare de la 16 biți la 8 biți), 626
conectori (elemente), 43-44
conectorul virgulă (,), 43
conexiuni (HTTP), 354
Connolly, Dan, 624
consecvența (proiectare), 461, 472, 475, 542-543
consorțiul Unicode, 626
consorțiul World Wide Web (W3C), 15
founding, 27-28
HTML 2, 26
sit Web, 14
constante, 871
construcții, titluri, 72
controale (formulare), 194-197
butoane image, 198
butoane radio, 200-201
butoane reset, 198, 205
butoane submit, 198, 205
casete de dialog, 198-199
dezactivare, 203-204
etichete, 208-209, 656
fișiere, 201-202
legende, 209-210, 656-657
navigare cu taste de acces, 211-212
navigarea tab, 211
seturile de câmpuri, 209-210
controlul pentru conștientizarea datelor (ActiveX), 443
conținut
conținut dinamic, definiția, 215
Modelul de Obiect Document (DOM), 396-397
tabele, inserare, 134-136, 139-140
vezi și text
conținut dinamic, definiția, 215
coordonate
ferestre de browser, 261
hărți imagine client, 183, 184
layere, 300-301
copierea șabloanelor, 61-63
copii (etichete HTML), 395-396
copyright, pagini Web, 77
Corel PhotoPaint, 507, 901
corp (documente HTML), 31-32
CSS, *vezi* pagini stil în cascadă (foi de stil în cascadă), fișiere .css, 232, 817
culori, 77-78
atribute
atributul background-color, 820
atributul color, 820
culori pentru compararea browserelor, 801
definire, 228
eticheta <BODY>, 17
foi de stil în cascadă (CSS), 298-300, 502
fundaluri, 77, 502, 820
hiperlegături
active, 502
legături vizitate, 502
statice, 502
intercalare, 504, 513
palete, 498
culorile browserului, 504
Forumul DisAbilities, 556-557
paletă de culori neprotejată, 504
paletă de culori protejată, 499-501
palete personalizate, 505-507
problema accesibilității, 646-647
text, 228, 502, 536
valori hexazecimale, 246, 250
determinare, 500
HTML, 502
triplet hexazecimal, 77-78
culori intercalate, 504, 513
cuvântul cheie ANCESTOR, 577, 835
cuvântul cheie ANY, 43
cuvântul cheie CDATA, 41, 45-46

cuvântul cheie CHILD, 576
cuvântul cheie CHILE, 835
cuvântul cheie DESCENDANT, 576, 835
cuvântul cheie DITTO, 577, 835
cuvântul cheie EMPTY, 43
cuvântul cheie FOLLOWING, 577, 835
cuvântul cheie FSIBLING, 577, 835
cuvântul cheie HERE, 577, 835
cuvântul cheie HTML, 577, 835
cuvântul cheie ID, 45-46, 577, 835
cuvântul cheie IGNORE, 46
cuvântul cheie INCLUDE, 46
cuvântul cheie NAME, 45-46
cuvântul cheie NMTOKEN, 45-46
cuvântul cheie NUMBER, 45-46
cuvântul cheie PRECEDING, 577, 835
cuvântul cheie PSIBLING, 577, 835
cuvântul cheie PUBLIC, 41
cuvântul cheie RCDATA, 43
cuvântul cheie ROOT, 576, 835
cuvinte cheie
  # FIXED, 45-46
  # IMPLIED, 46
  # PCDATA, 43-44
  # REQUIRED, 46
  ANCESTOR, 577, 835
  ANY, 43
  CDATA, 41, 45-46
  CHILD, 576, 835
  DESCENDANT, 576, 835
  DITTO, 577, 835
  EMPTY, 43
  FOLOWING, 577, 835
  FSIBLING, 577, 835
  HERE, 577, 835
  HTML, 577, 835
  ID, 45-46, 577, 835
  IGNORE, 46
  INCLUDE, 46
  NAME, 45-46
  NMTOKEN, 45-46
  NUMBER, 45-46
  PRECEDING, 577, 835
  PSIBLING, 577, 835
  PUBLIC, 41
  RCDATA, 43
  ROOT, 576, 835
Cyberdog, 886

D

date
  JavaScript, 322-323
date de caracter înlocuibile, 43
datele caracterului, 43
DD (formatul day - zi), 94
DeBabelizer, 507, 901
declararea solicitată a marcării (required markup declaration) (RMD), 567, 584, 828
declarația break, 865-866
declarația Call, 877-878
declarația comment, 865-866
declarația continue, 865-866
declarația Dim, 870, 875
declarația do while, 865-866
declarația Do...Loop, 877-878
declarația End, 875
declarația Erase, 870
declarația Exit, 875
declarația export, 865-866
declarația for, 865-866
declarația For...Next, 877-878
declarația function, 865-866, 875
declarația if...else, 867
declarația If...Then...Else, 877-878
declarația import, 867
declarația new, 867
declarația Option Explicit, 877-878
declarația Preserve, 871
declarația Randomize, 870
declarația ReDim, 871
declarația Rem, 871
declarația return, 867
declarația Select Case, 878-879
declarația Set, 871
declarația Sub, 875
declarația switch, 867
declarația this, 867
declarația UBound, 871
declarația var, 867
declarația while, 867
declarația While...Wend, 881
declarația with, 867
declarații
  atribute pentru elemente, 45-47
  break, 865-866
  Call, 877-878
  comment, 865-866
  continue, 865-866
  Dim, 870, 875
  do while, 865-866
  Do...Loop, 877-878
  End, 875
  Erase, 870
  Exit, 875
  export, 865-866
  for, 865-866
  For...Next, 877-878
  Function, 875
  function, 865-866
  HTML DTDs, 39-40
  if...else, 867
  If...Then...Else, 877-878
  import, 867
  label, 867
  new, 867
  Option Explicit, 877-878
  Preserve, 871
  Randomize, 870
  ReDim, 871
  Rem, 871
  return, 867
  Select Case, 881
  Set, 871
  SGML, 37
    identificare, 37-38
    parametrul BASESET, 40
    secțiunea CAPACITY, 37-38
    secțiunea CHARSET, 37-38
    secțiunea FEATURES, 38-39

  secțiunea SCOPE, 38-39
  secțiunea SYNTAX, 38-39
  standard, 37-38
Sub, 875
switch, 867
this, 867
Ubound, 871
var, 867
while, 867
While...Wend, 881
with, 867
XML
  atribute, 829
  declararea solicitată a marcării (RMD), 584
  documente, 566-567
  elemente, 829
  entități, 829
  notații, 831-832
declarații etichetate, 867
defilare
  cadre, 276-277
  evenimentul onscroll, 409
definire
  celule (tabele), 133
  culori, 228
  entități, 43-44
  scripturi, 79
definiții
  formatare, 90
definițiile tipului documentului, *vezi* DTDs, documente
degradare elegantă (Microsoft DHTML și browsere), 386-387
DejaNews, 350, 377
denumire (tabele), 124
depanatori (JavaScript), 327
deplasarea layerelor, 308-309
Design Lab, 904
DevEdge Newgroup FAQ, 327
Dewsnap, Don, 537
dezimțuire (fonturi), 537
DHTML (Dynamic HTML), 325, 414
  avantaje, 414-415
casete de dialog (JavaScript), 320
legarea datelor, 435
  creare, 438-439
  formulare de date, 546
scripturi de tratare a evenimentelor, 415
  animație în momentul încărcării, 418, 422
  comentarii, 433-434
  inserare în documente HTML, 416-418
  interfețe cu utilizatorul, 430-432
  prelucrare de formular, 422-424
  scripturile de acțiune a utilizatorului, 424-425
standarde, 379
  Microsoft, 380, 385-388
  Netscape, 379, 383-386
  W3C, 379-383
atributul left, 301
atributul top, 301
atributul visibility, 305
dimensiunea imaginilor, efect asupra timpului de afișare, 59
dimensionarea cadrelor, 276-277
dimensiunea absolută (atributul font-size), 819
dimensiunea în puncte (fonturi), 224, 226-228, 239-240
dimensiunea relativă (atributul font-size), 819
directiva de server Web Content-type, 358
directiva de server Web Location, 358
directiva de server Web, Status, 358
directoare, 661
  creare, 113
  Yahoo!, 661
disabilități, utilizatori cu nevoi speciale
  Blindness Resource Center, 933
problema accesibilității
  atributul align, 650-651
  audio, 658
  cadre, 657, 756-757
  compliance checker, 645
  culori, 646-647
  dispozitive, 642-643, 656-657
  documente numai pentru text, 645, 658-659
  emoticonuri, 646
  eticheta <BLOCKQUOTE>, 650
  eticheta <PRE>, 650
  formulare, 203-204, 656-657
  hărți imagine, 654-655
  hearing-impaired users, 643-644
  hiperlegături, 656
  imagini, 651-656
  Java, 658
  JavaScript, 658
  liste, 647-648
  machetă, 648
  marcarea logică, 644-645
  modalități alternative, 644-645
  principiile unitare de accesibilitate ale sitului Web, 645
  punctuație, 646
  tabele, 648-650
  tehnologii asistente, 643-644
  video, 658
situl Low Vision and Deaf/Blind, 933
Web Accessibility Initiative, 931
discuție
  cadre, 278-281
  Java și JavaScript, 322-323
  JavaScript și VRML, 322-323
documente
  documente HTML, 29
    antete, 70-71
    clasificare după motoarele de căutare, 75

comparații cu documente XML, 583
legare, 73-74
structură, 30-34
documente SGML, 37, 40
documente XML, 564, 595, 827
caractere, 827
comentarii, 827
comparații cu documente HTML, 583
declarare, 566-567
formatul corespunzător, 565
instrucțiuni de procesare, 828
prologuri, 828
Required Markup Declaration(RMD), 828
secțiunile CDATA, 828
spațiu alb, 828
structura fizică, 827
structura logică, 827
validare, 564, 590-591
foile de stil JavaScript, 289
Modelul de Obiect Document (DOM), 396-397
recenzii, adăugarea datelor la, 16, 94
listă glosar, 111
documente HTML, 29
anteturi, 70-71
aplicații CGI, apelare, 358-359
clasificare după motoare de căutare, 75
înglobare
scripturi JavaScript, 316-318
scripturi VBScript, 335-338
legare, 73-74
scripturi de tratare a evenimentelor, inserare, 416-418
structură, 30, 33
antet, 31-32
cadre, 31-33
corp, 31-32
foi de stil, 31
scripturi, 31
tabele, 33
documente interconectate (legături XML), 578, 835
domenii
numere în liste ordonate, 106
pointeri extinși (legături XML), 577
domeniu de existență, 402-403
domeniul de existență al subrutinei de tratare a evenimentului, definire, 402-403
DoodlePad, 324
Dortch, Bill, 324
dreptunghiuri, design considerations, 531
DSSL (limbaj de specificare a semanticii și stilului documentelor), 579
DSSSL (XSL), 600-604
DTDs (definițiile tipului documentului), 24, 35, 46
caracteristici nerecomandate, 46
declarația ATTLIST, 45
definire, 29
elemente, 42
entități, 39-40
entități generale, 41
entități parametru, 39-40
referiri la caractere, 41
HTML 4.0, 39
identificatori publici, 40-41
SGML, 37
URLs, 40
XML, 567
creare, 591-596
declararea elementelor, 567-568
declararea entităților, 570-571
declararea solicitată a marcării (required markup declaration) (RMD), 567, 828
declarațiile atributelor, 569-570
needs assessment, 590-591
secțiuni condiționale, 572, 830-831
secțiunile CDATA, 572
sintaxă, 828
subseturi externe, 567
subseturi interne, 567
XSL, 621
dublu clic (evenimentul ondblclick), 405
Dynamic Diagrams MAPA Service, 896
Dynamic HTML, vezi DHTML

E

e-mail, 24-25
adrese
formatare, 94
superlegături, 95
MIME, 628-629
Post Office Protocol 3 (POP3), 628
procesarea formularelor, 195
Internet Explorer, 195
Netscape Navigator 4.03, 195
Simple Mail Transfer Protocol (SMTP), 628
URL-urile mailto, 161-163
EasyDTD, 591
ECMA (Asociația Europeană pentru Standardizarea Informației și Sistemelor de Comunicații), 392, 579, 617-618
ECMAScript, 392, 579
exemple, 617-619
specificații, 617-618
utilizări, 617-618
ecrane (calculatoare)
setările rezoluției, 480
zona de proiectare, 480
ecrane de deschidere, 467, 473, 492
editarea textului
data și timpul editărilor, 16, 94

inserarea marcajului de text, 93
editoare (HTML), 887
Amaya, 890
BBEdit, 890
Composer, 889
Emacs, 893
functionalitate, 236
Home Page, 890
HomeSite, 890
Hot Dog Express, 889
Hot Dog Pro, 890
HoTMetaL Pro, 889
Multi-Edit, 891-892
PageMill, 890
selectare, 233
tabel de comparare, 888-889
Vi, 893
Visual SlickEdit, 891
Webber, 35, 891
WebEdit Pro, 891
editorul Multi-Edit, 891-892
ediții, adăugarea datelor la, 16, 94
efecte de margine (imagini), 510
elemente, 42
atribute, 45-47
atribute de aliniere, 68
atribute internaționale, 68
atributele core, 68
prestabiliri, 45
valori, 45-46
caracteristici nerecomandate, 46
conectori, 43-44
definirea, 29
elemente de frază, 81, 89-94, 115, 167, 716-719
elemente de text, 713-719
elemente goale, 43
elemente la nivel de bloc, 686-691
elemente matematice, 773
elemente speciale, 81, 115, 167
elemente structurale, 68, 80, 673-675
elementele antetului, 675-680
elementele cadrului, 681-684
elementele font, 81, 115, 167
elementele formularului 91, 116, 167, 194, 695-702
elementele inline, 115, 167
elementele listei, 691
elementele tabelului, 704-712
elementul STYLE, 31-32
elementul TABLE, 33
găzduire, 32
indicatori de apariție, 43-44
poziționare
elemente suprapuse, 268-269
foi de stil în cascadă, 261-271
poziționare absolută, 262-264
poziționare relativă, 261-262, 265-268
scripturi, 270-272
SGML, 37
XML
declarații, 829
DTDs, 567-568, 591-596
structuri fizice, 830
structuri logice, 830
vezi și etichete
elemente de conectare, 905
Acrobat reader, 173-174
inserare, 218-219
problemele utilizatorului, 218-219
ScriptActive (VBScript), 334
Shockwave, 218-219
VBScript, 392
elemente de frază, 81, 89-94, 115, 167, 716-719
elemente de mapare (XML DTDs), 591-592
elemente de tabel, 704-712
elemente inline, 115, 167
elemente la nivel de bloc, 686-691
elemente matematice, 773
elemente speciale, 81, 115, 167
elemente structurale, 68, 80, 673-675
elemente text, 713-719
elementele antetului, 675-680
elementele font, 81, 115, 167
elementele formularului, 82, 116, 167, 194, 695-702
elementele frame, 681-684
elementele listei, 691
Emacs, 893
Emacs W3, 800, 886
emoticonuri, probleme de accesibilitate, 646
Encyclopedia of Grafics File Formats, 905
Enquire, 22
entitatea document (XML), 831-832
entități, 39, 46
definire, 43-44
DTDs, 40
entitate HTML.Version (HTML 3.2), 40
entități escaped, 800
entități generale, 39-41
entități parametru, 39
entitățile caracter mnemonice &sky;, 638
entitățile caracter mnemonice &zwj, 637
entitățile caracter mnemonice &zwnj, 637
referiri la caractere, 41
XML, 565
declarații, 831-832
DTDs, 570-571, 594-595
entitatea document, 831-832
entități binare, 570, 594, 831
entități externe, 571
entități interne, 570
entități parametru, 570, 594
entități predefinite, 572, 832-833

entități text, 570, 594, 832
referințe, 831
reguli procesor XML, 832-833
structuri fizice, 830
structuri logice, 830
entități binare (XML), 570, 594, 597-598, 831
entități binare externe (XML), 597-598
entități escaped, 800
entități externe (XML), 571, 832
entități generale, 39-41
entități interne (XML), 570, 832
entități parametru
DTDs, 39
XML, 570, 594
entități predefinite (XML), 572, 832-833
entități text (XML), 570, 594, 832
entități text externe, 596-597
entități text interne, 596
entități text externe (XML), 596-597
entități text interne (XML), 596
entitățile caracter mnemonice
&sky, 638
&zwj, 637
&zwnj, 637
erori (elemente nerecomandate), 17
erori de programare și programul de depanare JavaScript, 326-327
eticheta </A>, 169, 722
eticheta </ACRONYM>, 717
eticheta </ADDRESS>, 686, 717
eticheta <A>, 169, 722
apelarea aplicațiilor CGI, 358
atributul href, 169-170
atributul id, 170
atributul name, 170
atributul rel, 169-170
atributul rev, 169-170
atributul target, 169-170
eticheta <ACRONYM>, 16, 90-91, 717
eticheta </APPLET>, 723
eticheta </B>, 713
eticheta </BDO>, 724
eticheta </BIG>, 714
eticheta </BLOCKQUOTE>, 686, 717
eticheta </BODY>, 673-674
eticheta </BUTTON>, 696, 732
eticheta </CAPTION>, 704-705
eticheta </CENTER>, 522, 687
eticheta </CITE>, 717
eticheta </CODE>, 718
eticheta </COLGROUP>, 705-706
eticheta </DD>, 691
eticheta </DEL>, 718
eticheta </DFN>, 719
eticheta </DIR>, 691-692
eticheta </DIV>, 520, 687-688
eticheta </DL>, 692
eticheta <!DOCTYPE>, 30, 68, 299, 685
eticheta </DT>, 693
eticheta </EM>, 719
eticheta </FIELDSET>, 696
eticheta </FONT>, 725
eticheta </FORM>, 697-698
eticheta </FRAME>, 683
eticheta </FRAMESET>, 682
eticheta </H1>, 688
eticheta </H6>, 688
eticheta </HEAD), 70-71, 674
eticheta </HTML>, 675
eticheta </I>, 714
eticheta </IFRAME>, 725
eticheta </INPUT>, 732
eticheta </INS>, 719-720
eticheta </ISINDEX>, 690
eticheta </KBD>, 720
eticheta </LABEL>, 699-700, 732
eticheta </LEGEND>, 700-701
eticheta </LI>, 101, 693-694
eticheta </MAP>, 731-732
eticheta </MENU>, 694-695
eticheta </NOFRAMES>, 684
eticheta </NOSCRIPT>, 432, 678, 726
eticheta </OBJECT>, 214, 449, 727-728
eticheta </OL>, 99, 694-695
eticheta </OPTION>, 701-702
eticheta </P>, 235, 779
eticheta </PLAINTEXT>, 780
eticheta </PRE>, 690, 720
eticheta </Q>, 729
eticheta </S>, 714
eticheta </SAMP>, 720-721
eticheta </SELECT>, 702, 732
eticheta </SMALL>, 715
eticheta </SPAN>, 729-730
eticheta </STRIKE>, 715
eticheta </STRONG>, 721
eticheta </STYLE>, 232, 680
eticheta </SUB>, 730
eticheta </SUP>, 730
eticheta </TABLE>, 116, 132-133, 706-707
eticheta </TBODY>, 707
eticheta </TEXTAREA>, 703-704, 732
eticheta </TFOOT>, 709-710
eticheta </TH>, 118, 710-711
eticheta </THEAD>, 711
eticheta </TITLE>, 680
eticheta </TR>, 712
eticheta </TT>, 716
eticheta </U>, 716
eticheta </UL>, 695
eticheta <ADDRESS>, 94, 96, 115, 36-37, 717
eticheta <ANCHOR>, 169-170
eticheta <APP>, 740

eticheta <APPLET>, 17, 154-155, 213, 216-217, 723
eticheta <AREA>, 181, 731
atributul accesskey, 184
atributul alt, 184, 654
atributul nohref, 184
atributul shape, 181-183
atributul tabindex, 184
atributul target, 184
eticheta <B>, 17, 89, 115, 223, 713
eticheta <BANNER>, 274
Eticheta <BASE>, 73, 159-160, 228, 675-676
atributul target, 280
eticheta end, 759
eticheta <BASEFONT>, 17, 723-724
eticheta <BDO>, 637, 724
eticheta <BGSOUND>, 220-221
eticheta <BIG>, 224, 714
eticheta <BLOCKQUOTE>, 95, 96, 518, 650, 686, 717
eticheta <BODY>, 31-32, 77-79, 673-674, 744
atributul bgcolor, 502
atributul leftmargin, 519
atributul link, 502
atributul text, 502-503
atributul topmargin, 519
color formation, 17
eticheta <BQ>, 746
eticheta <br clear>, 523
eticheta <BUTTON>, 16, 205, 696, 732
eticheta <CAPTION>, 124, 704-705
eticheta <CENTER>, 17, 78, 522, 687, 747
eticheta <CITE>, 90, 96, 717
eticheta <CODE>, 91, 97, 115, 718
eticheta <COL>, 125-127, 705
eticheta <COLGROUP>, 16, 125-127, 705-706
eticheta <COMMENT>, 750
eticheta <DD>, 111-112, 691
eticheta <DEL>, 16, 93-94, 225, 718
eticheta <DFN>, 90, 97, 719
eticheta <DIR>, 17, 111-113, 691-692
eticheta <DIV>, 78, 234, 270-272, 521-523, 687-688
atributul center, 521
atributul left, 521
atributul right, 521
eticheta <DL>, 111-112, 692
eticheta <DT>, 111-112, 693
eticheta <EM>, 32, 89, 97, 115, 223, 719
eticheta <EMBED>, 213, 218-219, 754
eticheta <FIELDSET>, 16, 209-210, 696
atributul align, 209-210
problema accesibilității, 656-657
eticheta <FONT>, 17, 226-228, 532, 725
atributul color, 536
atributul face, 534
atributul number, 226-228
atributul size, 226-228, 535
foi de stil, 17
interpretarea browserului, 757
eticheta <FORM>, 45, 194, 697-698, 767
atributul accept-charset, 196
atributul action, 195, 358
atributul enctype, 195-196
atributul method, 195
atributul onreset, 196
atributul onsubmit, 196
eticheta <FRAME>, 32, 274, 276-277, 281, 681
atributul frameborder, 277
atributul marginheight, 277
atributul marginwidth e, 277
atributul name, 276-277
atributul noresize, 276-277
atributul src, 276-277
atributul target, 278-280
elemente de accesibilitate, 758
motoare de căutare, 668
eticheta <FRAMESET>, 32, 275-277, 281, 682
browsere care nu suportă cadrele, 32
motoare de căutare, 668
eticheta <H1>, 32, 82, 688
eticheta <H6>, 82, 688
eticheta <HEAD>, 31, 70-71, 674, 759
eticheta <HR>, 96, 688, 689
eticheta <HTML>, 30, 69, 70, 675
eticheta <HYPH>, 638
eticheta <I>, 17, 89, 115, 223, 714
eticheta <IFRAME>, 282, 683, 725
atributul align, 282-283
atributul height, 282-283
atributul width, 282-283
înlocuirea cu <OBJECT>, 15
eticheta <ILAYER>, 764
eticheta <IMG>, 147, 154-155, 213, 726
apelarea aplicațiilor CGI, 358
atributul align, 148, 514, 522
atributul alt, 147-148, 215, 514, 652
atributul border, 151, 514
atributul controls, 220
atributul dynsrc, 220
atributul height, 150-151, 478, 514
atributul hspace, 151, 514, 525
atributul img, 513
atributul ismap, 152-153, 185
atributul left, 523
atributul loop, 220
atributul right, 523
atributul src, 147-148, 513
atributul start, 220
atributul usemap, 180

atributul vspace, 151, 514, 525
atributul width, 150-151, 478, 514
eticheta <INPUT>, 698-699, 732
atributul accesskey, 211-212
atributul alt, 203-204
atributul disabled, 203-204
atributul tabindex, 211
atributul type, 196-197
atributul usemap, 205
eticheta <INS>, 16, 93-94, 224, 719-720
eticheta <ISINDEX>, 17, 676, 690, 699-700, 767
atributul prompt, 205
deprecation status, 205
eticheta end, 759
eticheta <KBD>, 92, 97, 720
eticheta <LABEL>, 208-209, 699-700, 732
atributul for, 208-209
problema accesibilității, 656
eticheta <LAYER>, 270-272, 299, 764
eticheta <LEGEND>, 16, 209-210, 700-701
eticheta <LI>, 100-102, 693-694
eticheta <LINK>, 73-74, 676-677, 817
eticheta end, 759
Mosaic, 74
eticheta <LISTING>, 17, 88
eticheta <MAP>, 731-732
atributul name, 181
atributul usemap, 153
eticheta <MARQUE>, 772
eticheta <MENU>, 17, 113, 694-695
eticheta <META>, 75-77, 678
eticheta end, 759
mecanisme de căutare, 664, 667-669
eticheta <NEXTID>, 774
eticheta <NOEMBED>, 219
eticheta <NOFRAMES>, 274, 281-282, 668, 684, 758
eticheta <NOSCRIPT>, 432, 678, 726
eticheta <OBJECT>, 15, 153-154, 184-185, 213-214, 727-728, 754
atributul CLASSID, 438
atributul classid, 216-217
atributul classname, 216
atributul codebase, 216-217
atributul codetype, 216-217
atributul data, 214
atributul ID, 438
atributul shapes, 184-185
atributul type, 214
conștientizarea datelor, 449
legarea datelor 535-536
eticheta <OL>, 99, 694-695
atributul compact, 111
atributul start, 104-108
atributul type, 106-108
eticheta <OPTION>, 205-207, 701-702
eticheta <OVERLAY>, 778
eticheta <P>, 32, 42, 84, 235, 522, 690
atributul center, 522
atributul left, 522
atributul right, 522
eticheta <PARAM>, 154-155, 216, 728
atributul paramName, 216
atributul paramValue, 216
atributul type, 216
atributul valuetype, 216
eticheta <PLAINTEXT>, 17, 88
eticheta <PRE>, 17, 87-88, 690, 720
atributul width, 87
problema accesibilității, 650
eticheta <Q>, 15, 94-96, 637, 729
eticheta <S>, 17, 714
eticheta <SAMP>, 92, 97, 720-721
eticheta <SCRIPT>, 31-32, 79, 679, 729
atributul language, 317-318, 335
JavaScript, 317
Modelul Obiect pentru Scriptare Internet Explorer, 339
scripturi de tratare a evenimentelor 510, 432
VBScript, 335
eticheta <SELECT>, 205, 702, 732
atributul multiple, 206-207
atributul size, 206
eticheta <SMALL>, 224, 715
eticheta <SPAN>, 78, 729-730
eticheta <STRIKE>, 17, 224, 715
eticheta <STRONG>, 89, 97, 223, 721
eticheta <STYLE>, 31-32, 79, 232, 234, 680, 817
foile de stil în cascadă, 287
foile de stil JavaScript, 287
eticheta <SUB>, 225-226, 730
eticheta <SUP>, 225-226, 730
eticheta <TABLE>, 116, 132-133, 706-707
atribute, 116
atributul align, 116
atributul border, 116
atributul cellspacing, 117
atributul celpadding, 117
atributul cols, 116
atributul frame, 116
atributul rules, 117
atributul width, 116
eticheta <TBODY>, 124, 707
eticheta <TD>, 116, 118, 126, 133, 708-709
atributul align, 123
atributul axes, 123
atributul axis, 122
atributul colspans, 120
atributul nowrap, 124
atributul rowspan, 120
atributul valign, 123
atributul white-space, 124
eticheta <TEXTAREA>, 208, 703-704, 732
atributul cols, 208
atributul readonly, 208-209
atributul rows, 208
eticheta <TFOOT>, 124, 709-710
eticheta <TH>, 118-120, 710-711
eticheta <THEAD>, 124, 711
eticheta <TITLE>, 71-73, 680, 795
eticheta <TR>, 116-119, 133, 712
eticheta <TT>, 225, 716
eticheta <U>, 17, 223, 716
eticheta <UL>, 17, 108, 695
atributul compact, 111
atributul type, 109
eticheta <VAR>, 92, 97, 721
eticheta <-VAR>, 721
eticheta <XML>, 17, 88
etichete, 68
<! - - comment - ->, 737
<!DOCTYPE ... >, 685
<!DOCTYPE>, 30, 68, 299
<-DFN>, 719
</A>, 169, 722
</ACRONYM>, 717
</ADDRESS>, 686, 718
</AREA>, 181, 731
atributul accesskey, 184
atributul alt, 184, 654
atributul nohref, 184
atributul shape, 181-183
atributul tabindex, 184
atributul target, 184
</B>, 713
</BDO>, 724
</BIG>, 714
</BLOCKQUOTE>, 686, 717
</BODY>, 673-674
</BUTTON>, 696, 732
</CAPTION>, 704-705
</CENTER>, 522, 687-688
</CITE>, 717
</COLGROUP>, 705-706
</DD>, 691
</DEL>, 718
</DIR>, 691-692
</DIV>, 520, 687-688
</DL>, 692
</DT>, 693
</EM>, 719
</FIELDSET>, 696
</FONT>, 725
</FORM>, 697-698
</FRAME>, 683
</FRAMESET>, 682
</H1>, 688
</H6>, 688
</HEAD>, 70-71, 674
</HTML>, 675
</I>, 714
</IFRAME>, 725
</INPUT>, 732
</INS>, 719-720
</ISINDEX>, 690
</KBD>, 720
</LABEL>, 699-700, 732
</LEGEND>, 700-701
</LI>, 101, 693-694
</MAP>, 731-732
</MENU>, 694-695
</NOFRAMES>, 684
</NOSCRIPT>, 432, 678, 726
</OBJECT>, 214, 727-728
conștientizarea datelor, 449
legarea datelor, 438-439
</OL>, 99, 694-695
</OPTION>, 701-702
</P>, 235, 779
</PLAINTEXT>, 780
</PRE>, 690, 720
</Q>, 729
</S>, 714
</SAMP>, 720-721
</SCRIPT>, 680
Modelul Obiect pentru Scriptare Internet Explorer, 339
scripturi pentru tratarea evenimentelor, 416, 432
VBScript, 335
</SELECT>, 702, 732
</SMALL>, 224, 715
</SPAN>, 729-730
</STRONG>, 721
</STYLE>, 232, 680
</SUB>, 730
</SUP>, 730
</TABLE>, 116, 132-133, 706-707
</TBODY>, 707
</TD>, 118, 708-709
</TEXTAREA>, 703-704, 732
</TFOOT>, 709-710
</TH>, 118, 710-711
</THEAD>, 711
</TITLE>, 680
</TR>, 712
</TT>, 716
</U>, 716
</UL>, 695
</VAR>, 721
<A>, 169, 722
<ACRONYM>, 16, 90-91, 717
<ADDRESS>, 94, 96, 115, 686, 717
<APP>, 740
<APPLET>, 17, 154-155, 213, 216-217, 723
<B>, 17, 89, 115, 223, 713
<BANNER>, 274
<BASE>, 73, 159-160, 675-676
atributul target, 280
eticheta end, 759
<BASEFONT>, 17, 228, 723-724
<BDO>, 637, 724
<BGSOUND>, 220-221
<BIG>, 224, 714
<BLOCKQUOTE>, 95, 9(
518, 650, 686, 717

<BODY>, 17, 77-79, 673-674, 744
atributul bgcolor, 502
atributul leftmargin, 519
atributul link, 502
atributul text, 502-503
atributul topmargin, 519
<BQ>, 746
<br clear>, 523

, 85-86, 724-725
<BUTTON>, 16, 205, 696, 732
<CAPTION>, 124, 704-705
<CENTER>, 78, 522, 687-688, 747
<CITE>, 90, 96, 717
<CODE>, 91, 97, 115, 718
<COL>, 125-127, 705
<COLGROUP>, 16, 125-127, 705-706
<COMMENT>, 750
<DD>, 111-112, 691
<DEL>, 16, 93-94, 225, 718
<DFN>, 90, 97, 719
<DIR>, 17, 113, 691-692
<DIV>, 78, 234, 270-272, 521-523, 687-688
<DL>, 111-112, 692
<DT>, 111-112, 693
<EM>, 89, 97, 115, 223, 719
<EMBED>, 213, 218-219, 754
<FIELDSET>, 16, 209-210, 696
atributul align, 209-210
problema accesibilității, 656-657
<FONT>, 17, 226-228, 532, 725
atributul color, 536
atributul face, 534
atributul number, 226-228
atributul size, 226-228, 535
interpretarea browserului, 757
<FORM>, 194, 697-698, 767
atributul accept-charset, 196
atributul action, 195, 358
atributul enctype, 195-196
atributul method, 195
atributul onreset, 196
atributul onsubmit, 196
<FRAME>, 274, 276-277, 281, 681
atributul frameborder, 277
atributul marginwidth, 277
atributul name, 276-277
atributul noresize, 276-277
atributul scrolling, 276-277
atributul src, 276-277
atributul target, 278-280
motoare de căutare, 668
problema accesibilității, 656-657
<FRAMESET>, 275-277, 281, 682
evenimentul onLoad, 275
evenimentul onUnload, 275
<H1>, 82, 688
<H6>, 82, 688
<HEAD>, 70-71, 674, 759
<HR>, 96, 688
<HTML>, 30, 69-70, 675
<HYPH>, 638
<I>, 17, 89, 223, 714
<IFRAME>, 282, 683, 725
atributul align, 282-283
atributul height, 282-283
atributul width, 282-283
înlocuirea cu <OBJECT>, 15
<ILAYER>, 764
<IMG>, 147, 154-155, 213, 726
apelarea aplicațiilor CGI, 358
atributul align, 148, 514, 522
atributul alt, 147-148, 215, 514, 652
atributul border, 151, 514
atributul controls, 220
atributul dynsrc, 220
atributul height, 150-151, 478, 514
atributul hspace, 151, 514, 525
atributul img, 513
atributul ismap, 152-153, 185
atributul left, 523
atributul loop, 220
atributul right, 523
atributul src, 147-148, 513
atributul start, 220
atributul usemap, 180
atributul vspace, 151, 514, 525
atributul width, 150-151, 478, 514
<INPUT>, 698-699, 732
atributul accesskey, 211-212
atributul alt, 203-204
atributul disabled, 203-204
atributul tabindex, 211
atributul type, 196-197
atributul usemap, 205
<INS>, 16, 93-94, 224, 719-720
<ISINDEX>, 17, 676, 690, 699-700, 767
atributul prompt, 205
deprecation status, 205
eticheta end, 759
<KBD>, 92, 97, 720
<LABEL>, 208-209, 699-700, 732
atributul for, 208-209
problema accesibilității, 656
<LAYER>, 270-272, 299, 764
atributul left, 301
atributul top, 301
atributul visibility, 305
DHTML, 379
<LEGEND>, 16, 209-210, 700-701
<LI>, 100-102, 693-694

<LINK>, 73-74, 676-677, 817
eticheta end, 759
Mosaic, 74
<LISTING>, 17, 88
<MAP>, 731-732
atributul name, 181
atributul usemap, 153
<MARQUE>, 772
<MENU>, 17, 113, 694-695
<META>, 75-77, 678
eticheta end, 759
motoare de căutare, 664, 667-669
<NEXTID>, 774
<NOEMBED>, 219
<NOFRAMES>, 274, 281-282, 684, 758
<NOSCRIPT>, 432, 678, 726
<OBJECT>, 15, 153-154, 184-185, 213-214, 727-728, 754
atributul classid, 216-217, 438
atributul classname, 216
atributul codebase, 216-217
atributul codetype, 216-217
atributul data, 214
atributul ID, 438
atributul shapes, 184-185
atributul type, 214
conștientizarea datelor, 449
legarea datelor, 438-439
<OL>, 99, 694-695
atributul compact, 111
atributul start, 104-108
atributul type, 106-108
<OPTION>, 205-207, 701-702
<OVERLAY>, 778
<P>, 84, 235, 522, 690
atributul center, 522
atributul left, 522
<PARAM>, 154-155, 216, 728
atributul paramName, 216
atributul paramValue, 216
atributul type, 216
atributul valuetype, 216
<PLAINTEXT>, 17, 88
<PRE>, 87-88, 690, 720
atributul width, 87
problema accesibilității, 650
<Q>, 15, 94-96, 637, 729
<S>, 714
<SAMP>, 92, 97, 720-721
<SCRIPT>, 79, 679, 729
atributul language, 317-318, 335
JavaScript, 317
Modelul Obiect pentru Scriptare Internet Explorer, 339
scripturi pentru tratarea evenimentelor, 416, 432
VBScript, 335
<SELECT>, 205, 702, 732
atributul multiple, 206-207
atributul size, 206
<SMALL>, 224, 715
<SPAN>, 78, 729-730
<STRIKE>, 17, 224, 715
<STRIKE>, 715
<STRONG>, 89, 97, 223, 721
<STYLE>, 79, 232, 234, 680, 817
<SUB>, 225-226, 730
<SUP>, 225-226, 730
<TABLE>, 116, 132-133, 706-707
atribute, 116
atributul align, 116
atributul border, 116
atributul cols, 116
atributul frame, 116
atributul width, 116
<TBODY>, 124, 707
<TD>, 116, 118, 126, 133, 708-709
atributul align, 123
atributul axes, 123
atributul axis, 123
atributul colspans, 120
atributul nowrap, 124
atributul rowspan, 120
atributul valign, 123
atributul white-space, 124
<TEXTAREA>, 208, 703-704, 732
atributul cols, 208
atributul readonly, 208-209
atributul rows, 208
<TFOOT>, 124, 709-710
<TH>, 118-120, 710-711
<THEAD>, 124
<TITLE>, 71-73, 680, 795
<TR>, 116-119, 133, 712
<TT>, 225, 716
<U>, 17, 223, 716
<UL>, 17, 108, 695
atributul compact, 111
atributul type, 109
<VAR>, 92, 97, 721
apelarea aplicațiilor CGI, 358
atributul href, 169-170
atributul id, 170
atributul name, 170
atributul rel, 169-170
atributul rev, 169-170
atributul target, 169-170
clase, atribuire, 248-249
copii, 395-396
definiția, 30
elemente nerecomandate, 17
eticheta <! - - ... - ->, 684
eticheta <XML>, 17, 88
etichete de înfățișare, 17
etichete de proprietar, 39
etichete nerecomandate, 671
etichetele speciale SGML, 684
foile de stil în cascadă, 287
foile de stil JavaScript, 287

folosirea etichetelor multiple pentru a crea spațiu alb, 745
etichete (controale formulare), 208-209, 656
etichete de aspect
elemente nerecomandate, 17
eticheta <B>, 17
eticheta <BASEFONT>, 17
eticheta <CENTER>, 17
eticheta <FONT>, 17
eticheta <I>, 17
eticheta <STRIKE>, 17
eticheta <U>, 17
etichete de proprietar, 39
etichete de sfârșit
omiterea, 42
XML, 566, 829
etichetele start
omitere, 42
XML, 566, 829
etichetele structure, 16
eticheta <ACRONYM>, 16
eticheta <BUTTON>, 16
eticheta <COLGROUP>, 16
eticheta <DEL>, 16
eticheta <FIELDSET>, 16
eticheta <INS>, 16
eticheta <LEGEND>, 16
eticheta <Q>, 16
omitere
etichete de început, 42
etichete de încheiere, 42
SGML, 38-39
părinți, 395-396
problema accesibilității, 645
scripturi, folosirea etichetelor HTML, 417-418
selectori, 816
tabelul de comparare a browserelor, 736-880
XML
etichete de început, 566, 829
etichete de sfârșit (etichete de încheiere), 566, 829
structuri fizice, 831
structuri logice, 831
European Laboratory for Particle Physics (CERN), 22, 36, 185-186
evenimente, 45
barbotarea evenimentului, (agitarea evenimentului), 397, 403, 735
comportarea browserului, 735
evenimente intrinseci, 399, 733, 799
evenimente programate (Forumul DisAbilities), 549
evenimentul onabort, 409
evenimentul onafterupdate, 409
evenimentul onbeforeunload, 409
evenimentul onbeforeupdate, 409
evenimentul onblur, 407
evenimentul onchange, 408
evenimentul onclick, 405
evenimentul ondatasetavailable, 409
evenimentul ondatasetcomplete, 409
evenimentul ondblclick, 405
evenimentul onerrorupdate, 409
evenimentul onfinish, 409
evenimentul onfocus, 407
evenimentul onhelp, 409
evenimentul onkeydown, 407
evenimentul onkeypress, 407
evenimentul onkeyup, 407
evenimentul onload, 275, 404
evenimentul onmousedown, 405
evenimentul onmousemove, 406
evenimentul onmouseout, 406
evenimentul onmouseover, 406
evenimentul onmouseup, 405
evenimentul onreset, 196, 408
evenimentul onrowenter, 409
evenimentul onrowexit, 409
evenimentul onscroll, 409
evenimentul onselect, 408
evenimentul onstart, 409
evenimentul onsubmit, 196, 408
evenimentul onunload, 275, 404
funcționalitate, 400-402
evenimente de la tastatură
evenimentul onblur, 407
evenimentul onfocus, 407
evenimentul onkeydown, 407
evenimentul onkeypress, 407
evenimentul onkeyup, 407
evenimente intrinseci, 399, 733, 799
evenimente pentru elemente de indicare
evenimentul onblur, 656
evenimentul onclick, 405
evenimentul ondblclick, 405
evenimentul onfocus, 407
evenimentul onmousedown, 405
evenimentul onmousemove, 406
evenimentul onmouseout, 406
evenimentul onmouseover, 406
evenimentul onmouseup, 405

evenimentele mouse
evenimentul onblur, 407
evenimentul onclick, 405
evenimentul ondblclick, 405
evenimentul onfocus, 407
evenimentul onmousedown, 405
evenimentul onmousemove, 406
evenimentul onmouseout, 406
evenimentul onmouseover, 406
evenimentul onmouseup, 405
evenimentele programate (forumul DisAbilities), 549
Modelul de Obiect Document (DOM), 394
rutină de ascultare a evenimentelor, 400-401
scripturi de tratare a evenimentelor, 415
animație în momentul încărcării, 418, 422
browser support, 432-433
comentarii, 433-434
inserare în documente HTML, 416-418
interfețe cu utilizatorul, 430-432
limbaje de scriptare, 415
prelucrare de formular, 422-424
scripturile de acțiune a utilizatorului, 424-425
standarde
scopul, 400
W3C, 399
subrutine de tratare a evenimentelor, 68, 400
domeniu de existență, definire, 402-403
evenimentul onload, 77-78, 338
evenimentul onunload, 77-78
inline event handlers, 401, 409-412
Internet Explorer, 409
JavaScript, 317
obiectul area, 838
obiectul button, 838
obiectul checkbox, 839
obiectul combo, 840
obiectul document, 842
obiectul form, 843
obiectul frame, 845
obiectul image, 846
obiectul layer, 848
obiectul link, 849-850
obiectul password, 853
obiectul radio, 854
obiectul reset, 856
obiectul select, 857
obiectul submit, 859-860
obiectul text, 861
obiectul textarea, 861-862
obiectul window, 865
evenimentul onabort, 409
evenimentul onafterupdate, 409
evenimentul onbeforeunload, 409
evenimentul onbeforeupdate, 409
evenimentul onblur, 407
evenimentul onchange, 408
evenimentul onclick, 405, 839
evenimentul ondatasetavailable, 409
evenimentul ondatasetcomplete, 409
evenimentul ondblclick, 405
evenimentul onerrorupdate, 409
evenimentul onfinish, 409
evenimentul onfocus, 407, 839
evenimentul onhelp, 409
evenimentul onkeydown, 407
evenimentul onkeypress, 407
evenimentul onkeyup, 407
evenimentul onLoad, 77-78, 404
eticheta <FRAMESET>, 275
VBScript, 338
evenimentul onmousedown, 405, 839
evenimentul onmousemove, 406
evenimentul onmouseout, 406
evenimentul onmouseover, 406
evenimentul onmouseup, 405, 839
evenimentul onreset, 196, 408, 843
evenimentul onrowenter, 409
evenimentul onrowexit, 409
evenimentul onscroll, 409
evenimentul onselect, 408
evenimentul onstart, 409
evenimentul onsubmit, 196, 408, 843
evenimentul onunload, 77-78, 275, 404
Excite, 350, 361, 375-377, 663, 942
eticheta <META>, 668
ExperTelligence Webber 32, 891
ExperTelligence WebberActive, 895
explicarea paginilor Web, 77
Explicit association of labels with form controls, 208-209
Explorer, 338-340
Explorer, *vezi* Internet Explorer
Extensible Markup Language, *vezi* XML
Extensible Style Language, *vezi* XSL
extensiile fișierelor și browsere, 213-214
extranet, definiție, 465

F

filozofia limbajului HTML, 671
filtrarea formularelor de date, 438
filtre
imagini, 510
fișiere
fișiere .css, 232, 817
formulare, încărcare, 201-202
salvare, 513
fișiere GIF, 512
fișiere JPG, 512
URL-uri, 164
fișiere .avi
cadre, 220
inserare, 220
începerea, 220
fișiere .dcr, 213
fișiere .js, 318
fișiere .map, 185
fișiere .pdf, 841
fișiere .ram, 174
fișiere .rpm, 174
fișiere .wrl, 213
fișiere Acrobat, hiperlegături la, 173-174
fișiere audio
hiperlegături, 173-174
inserare, 220
problema accesibilității, 658
fișiere audio RealAudio, 173-174, 906, 909
software, 909
fișiere JPEG, 513
fișiere video
bucle, 220
cadre, 220
inserare, 214, 220
pornire, 220
problema accesibilității, 658
fișierele JPG progresive, 513
fișierul film QuickTime, inserare, 214
fișierul tip format de valori separate prin virgule (CSV), 443
Flash (Macromedia), 487, 549
Flavell, Alan, 652
foi de stil
CSS, 230, 816
atributul textalign, 287
combinarea cu foile de stil JavaScript, 296-298
culoare, 245-247, 502
eticheta <STYLE>, 287
margini, 241-243
Netscape, 287
paginile de stiluri inline, 817
paginile de stiluri înglobate, 817
paginile de stiluri legate, 817
poziționarea elementului, 261-271
reguli pentru stil, 817-818
sintaxa, 238, 816
W3C, 230
XML, 578, 587-588
XSL, 600
foi de stil în cascadă (Cascading Style Sheets)
foile de stil JavaScript (JSSS), 287
atributul textalign, 287
clase, 288
combinarea cu foile de stil în cascadă, 296-298
documente, 289
eticheta <STYLE>, 287
etichete, 288
foi de stil externe, 296
foile de stil în cascadă, 287-288
IDs, 288
layere, 309-314
metoda contextual, 289
Netscape, 287
pagini Web, 290-296
proprietatea apply, 289-291
standarde W3C, 288
stiluri condiționale, 289
suportul browserului, 222, 523
tabele, 116
text, 222
XSL, 577-578, 600
anularea, 616
DSSL, 579, 600-601
ECMAScript, 579, 617-619
foi de stil cu structură fixă, 620-621
foi de stil în cascadă (CSS), 600
functionality, 579, 601
importarea, 616
obiectele de curs, 601-605
procesarea, 579-580
reguli de construcție, 579, 605-613
reguli de stil, 606, 614-615
XML DTDs, 621
foi de stil externe, 232, 236-237
foile de stil JavaScript, 296
sintaxă, 232
foi de stil inline, 232-235
mixarea tehnicilor foii de stil, 238
foi de stil în cascadă (CSS), 230, 816
aliniere (text), 244-245
atribuirea claselor, 248-249
atribute
atributul background, 821
atributul background-attachment, 821
atributul background-color, 820
atributul background-image, 820
atributul background-position, 821
atributul background-repeat, 821
atributul border, 825

atributul border-bottom, 823
atributul border-bottom-width, 824
atributul border-color, 824
atributul border-left, 823
atributul border-left-width, 824
atributul border-right, 823
atributul border-right-width, 824
atributul border-style, 824
atributul border-top, 823
atributul border-top-width, 824
atributul border-width, 824
atributul clear, 825
atributul color, 820
atributul display, 826
atributul float, 825
atributul font, 820
atributul font-family, 818
atributul font-size, 819
atributul font-style, 819
atributul font-variant, 819
atributul font-weight, 819
atributul height, 825
atributul letter-spacing, 821
atributul line-height, 822
atributul list-style, 826
atributul list-style-image, 826
atributul list-style-position, 826
atributul list-style-type, 826
atributul margin, 823
atributul padding, 823
atributul padding-bottom, 823
atributul padding-left, 823
atributul padding-right, 823
atributul padding-top, 823
atributul text-align, 287, 822
atributul text-decoration, 821
atributul text-transform, 822
atributul vertical-align, 822
atributul white-space, 826
atributul width, 825
atributul word-spacing, 821
browsere, 230
combinarea cu foile de stil JavaScript, 296-298
culoare, 245-247, 502
culoarea din fundal, 245-247
definiții
paginile stil inline, 817
paginile stil înglobate, 817
paginile stil legate, 817
sintaxă, 816
elemente de poziționare, 261
elemente suprapuse, 268-269
poziționare absolută, 262-264
poziționare relativă, 265-268
poziționarea relativă contra poziționării absolute, 261-262
scripturi, 270-272
eticheta <STYLE>, 287
foi de stil inline, 232-235
foile de stil JavaScript, 287
functionalitate, 230
grupare, 247
limitate, 231-232
margini, 241-243
Netscape, 287
reguli referitoare la stil, 817-818
resurse online, 930
sintaxa, 238
șabloane, 249-250
W3C, 230
XML, 578, 587-588
XSL, 600
foi de stil înglobate, 232, 234-236
Cascading Style Sheets(CSS), 817
sintaxă, 232
foi de stil JavaScript (JSSS), 287
atribute, 287
clase, 288
combinarea cu foile de stil în cascadă, 296-298
documente, 289
eticheta <STYLE>, 287
etichete, 288
foi de stil externe, 296
foile de stil în cascadă, 287
IDs, 288
layere, 309-314
metoda contextual, 289
Netscape, 287
pagini Web, 290-296
proprietatea apply, 289-291
standarde W3C, 288
stiluri condiționale, 289
foi de stil legate, 232, 236-237
Cascading Style Sheets(CSS), 817
mixarea tehnicilor foii de stil, 238
sintaxă, 232
folosirea marginilor din dreapta pentru navigare, 489-491
cadre, 491
tabele, 489-491
folosirea marginilor din stânga pentru navigare, 489-491
cadre, 491
tabele, 489-491
folosirea părții de jos a paginilor Web pentru navigare, 488
cadre, 488
tabele, 488

fontul Cyberbit, 639
fonturi, 226-228
amestecare, 534
atribute, 818
atributul font, 820
atributul font-family, 238-240, 818
atributul font-size, 239-240, 819
atributul font-style, 240, 819
atributul font-variant, 819
atributul font-weight, 240, 819
culori, 536
design, 532-537, 537
fancy, 818
foi de stil în cascadă (CSS), 238-242
fontul Cyberbit, 639
fonturi cu serife, 239, 532-533, 818
fonturi cursive, 239, 818
fonturi decorative, 240, 532-533
fonturi fără șerife, 239, 532-533, 818
fonturi încapsulate, 535
fonturi monospațiate, 239, 532-533, 818
fonturile fantasy, 239
fonturile script, 532-533
grosimi, 240
internaționalizarea limbajului HTML, 639-641
încapsulare, 535, 639-640
mărimi, 226-228, 239-240, 535-537
netezire, 537
prestabiliri, 535
readability, 239
resurse, 239
stiluri, 240
suportul browserului, 639
fonturi cu serife, 239, 532-533, 818
fonturi cursive, 239, 532-533, 818
fonturi de manuscris, 239, 532-533
fonturi decorative, 239, 532-533
fonturi fancy, 818
fonturi fantasy, 239
fonturi fără serife, 239, 532-533, 818
fonturi neșlefuite, 537
fonturi netezite, 537
FORMAL YES, 38-39
formatare
numere în tabele, 118
tabele, 116
text, 222
acronime, 90
adrese, 94
anteturi, 82
barat, 224
bold, 223
citate, 94-96
cod, 91-93
culori, 228
definiții, 90
exponent, 225-226
fonturi, 226-228
indice, 225-226
italic, 223
întreruperi de rând, 85-87
marcaje pentru editare, 93-94
marcarea textului inserat, 224
mărime, 224
monospațiat, 91-93, 225
paragrafe, 84-85
problema accesibilitățtii, 645
subliniat, 223-224
text estompat, 93
text evidențiat, 89
text inserat, 93
text preformatat, 87-88
formatul corespunzător (buna-formare)
formatul UTF-8 (ISO 10646), 633
forme, 530-531
design, 531
forumul DisAbilities, 560-562
hărți de imagine
partea clientului, 181-183
partea serverului, 185
formulare, 24-25, 194
casetă cu listă, 205-207
CGI
procesarea datelor, 357
rutine de tratare a formularelor, 358, 363, 368-370
validarea datelor, 343
codificare, 195-196
controale, 194-197
butoane image, 198
butoane push, 197, 205
butoane radio, 200-201
butoane reset, 198, 205
butoane submit, 198, 205
casete de dialog, 198-199
dezactivare, 203-204
etichete, 208-209, 656
fișiere, 201-202
legende, 209-210, 656-657
navigarea cu taste de acces, 211-212
navigarea tab, 211
parole, 203-204
seturile de câmpuri, 209-210
DHTML, prelucrare, 422-424
evenimente
evenimentul onreset, 408
evenimentul onsubmit, 408
formulare de date, 438
actualizare, 546
câmpuri, 439
creare, 439
editarea înregistrărilor, 452
filtrare, 438
înregistrări, 438, 452
vizualizare înregistrări, 452
găzduire, 45
hărți imagine, 205
input, 196
meniu care se desfășoară, 205-207

problema accesibilității, 656
atributul <LABEL>, 656
atributul <LEGEND>, 656-657
eticheta <FIELDSET>, 656-657
navigarea cu ajutorul tastaturii, 656-657
procesare, 194-195
resetare, 196
setul de caractere
internaționalizare, 631-633
specificare, 196
suportul browserului, 203-204
text, zone, 208
text, casete, 208-209
transmitere, 196
validarea datelor (JavaScript), 319
validarea datelor VBScript, 340-343
formulare de date, 438
actualizare, 546
creare, 439
câmpuri, 439
filtrare, 438
înregistrări, 438
editare, 452
viewing, 452
formulare de resetare, 196
Forumul DisAbilities, 539
BBSs, 547-548
camere de discuții, 547-548
claritate, 541
consecvență, 542-543
conținut, 544-545
culori, 556-557
evenimentele programate, 549
experiența publicului, 546-547
forme, 560-562
imagini, 556-557
interfața, 539-541
metaforă, 541
multimedia, 548
navigare, 543, 549-556
orientare, 543
scopul proiectantului, 544
scopuri, 546
spațiul, 558-560
typography, 560-562
forumul DisAbilities
Microsoft Network, 539
BBSs, 547-548
camere de discuții, 547-548
claritate, 541
consistență, 542-543
conținut, 544-545
culori, 556-557
evenimentele programate, 549
experiența publicului, 546-547
forme, 560-562
imagini, 556-557
interfața, 539-541
metaforă, 541
multimedia, 548
navigare, 549-556
orientare, 543
scopul proiectantului, 544
scopuri, 546
spațiu, 558-560
tipografie, 560-562
fotografii (stoc de fotografii), 508
Fractal Design Painter, 901
Fragmente de URL-uri, 160-161
Freehand Graphics Studio, 902
FrontPage, 895
FTP (file transfer protocol), 24-25, 161, 164
funcția Abs, 871
funcția Asc, 871
funcția Atn, 870
funcția Cbool, 871
funcția Cbyte, 871
funcția Cdate, 871
funcția CDbl, 871
funcția Chr, 871
funcția CInt, 871
funcția CLng, 871
funcția CLTrim, 878-879
funcția Cos, 870
funcția CSng, 871
funcția CStr, 871
funcția Date, 873
funcția DateSerial, 873-874
funcția DateValue, 873-874
funcția Day, 874
funcția escape( ), 865
funcția eval( ), 865
funcția Exp, 870
funcția Fix, 873
funcția Hex, 873
funcția Hour, 875
funcția InputBox, 876
funcția InStr, 878-879
funcția Int, 873
funcția IsArray, 870, 879-880
funcția IsDate, 879-880
funcția IsEmpty, 879-880
funcția isNaN( ), 865
funcția IsNull, 879-880
funcția IsNumeric, 879-880
funcția LBound, 870
funcția LCase, 878-879
funcția Left, 878-879
funcția Len, 878-879
funcția Log, 870
funcția Mid, 878-879
funcția Minute, 875
funcția Month, 875
funcția MsgBox, 876
funcția Now, 875
funcția Oct, 873
funcția parseFloat( ), 865-866
funcția parseInt( ), 865-866
funcția Right, 878-879
funcția Rnd, 878-879
funcția RTrim, 878-879
funcția Second, 875
funcția Sgn, 873
funcția Sin, 870
funcția Space, 879-880
funcția Split (VBScript), 334
funcția Sqr, 870
funcția StrComp, 879-880
funcția String, 879-880
funcția Tan, 870
funcția Time, 875
funcția TimeSerial, 873-874
funcția TimeValue, 873-874
funcția toString( ), 865-866

funcția Trim, 879-880
funcția Ucase, 879-880
funcția unescape( ), 865-866
funcția VarType, 879-880
funcția Weekday, 875
funcția Year, 875
funcții
Abs, 871
Asc, 871
Atn, 870
CBool, 871
CByte, 871
CDate, 871
CDbl, 871
Chr, 872
CInt, 872
CLng, 872
Cos, 870
CSng, 872
CStr, 872
Date, 873
DateSerial, 873-874
DateValue, 873-874
Day, 874
escape( ), 865
eval( ), 865
Exp, 870
Fix, 873
funcția Split (VBScript), 334
Hex, 873
Hour, 874
InputBox, 876
InStr, 878-879
Int, 873
IsArray, 870, 879-880
IsDate, 879-880
IsEmpty, 879-880
isNaN( ), 865
IsNull, 879-880
IsNumeric, 879-880
LBound, 870
LCase, 878-879
Left, 878-879
Len, 878-879
Log, 870
LTrim, 878-879
Mid, 878-879
Minute, 874
Month, 874
MsgBox, 876
Now, 874
Oct, 873
parseFloat( ), 865-866
parseInt( ), 865-866
Right, 878-879
Rnd, 870
RTrim, 878-879
Second, 874
Sgn, 873
Sin, 870
Space, 879-880
Sqr, 870
StrComp, 879-880
String, 879-880
Tan, 870
Time, 874
TimeSerial, 873-874
TimeValue, 873-874
toString( ), 865-866
Trim, 879-880
UCase, 879-880
unescape( ), 865-866
VarType, 879-880
Weekday, 874
Year, 875
funcții conversion (VBScript), 872-873
funcții dată/oră (VBScript), 873-875
funcții independente (JavaScript), 865
fundaluri
atribute
atributul background, 821
atributul background-attachment, 821
atributul background-color, 820
atributul background-image, 820
atributul background-position, 821
atributul background-repeat, 821
culoare, 77, 502, 245-247, 820
imagini, 820
plasare, 821
repetare, 821
layere, 302
pagini Web, 491
Fusion, 893-894
fuzionarea celulelor (tabele), 120-122

G

Gamelan, 912
gateways, vezi CGI (Common Gateway Interface)
gazde
căi, 161-162
URL-uri, 156
găzduire
cadre, 275, 280, 285
elemente, 32
formulare, 45
liste, 106-107, 109
Gerken, Wil, 492
ghid de tipărire ușoară folosind Desktop Publisher, 537
ghilimele
citate inline, 15
design, 518
interpretarea spațiului alb de către browser, 743
formatare, 94-96
ghilimele și internaționalizarea limbajului HTML, 637
GIF Construction Set, 902
GIFBuilder, 220
GIFTool, 903
GIFTrans, 903
GIMP (Gnu Image Manipulation Program), 902
glife, 624-625
modul de scriere cursiv, 637
servere, 639
GML (limbajul generalizat de marcare), 36
GMT (Greenwich Mean Time), 94
GoldWave, 909
Gopher, 24-25

grafică, HTML+, 25
vezi și imagini
graficienii profesioniști, 505-506
Graphics Interchange Format, *vezi* imaginile GIF
Greenwich Mean Time (GMT), 94
grosimea liniilor de separare orizontală, 96
grupare
coloane (tabele), 16, 125-127
foi de stil în cascadă (CSS), 247
rânduri (tabele), 124
grupuri de discuții, 939
CGI, 377
DevEdge Newsgroup, 327
NNTP URLs, 163
URL-urile news, 161, 163
VBScript, 350
grupuri de legături extinse (legături XML), 578, 835
grupurile de discuții Usenet, 163, 939
Guitar Chord Computer, 320

H

Haifa Research Institute, 912
Haley Allan, 537
handlerul htimage CERN pentru harta imagine, 186
handlerul imagemap NCSA pentru harta imagine, 186
harta de sit Exploder, 897
hărți, 176
browsere nongrafice, 184
browserul Lynx, 741, 771
CLEARWeb, 896
Exploder Site Map, 897
Governor Site MAp, 897
HotSauce, 897
Joe Bart's Site Map, 897
MAPA Service, 896
MapXsite, 896
SiteAnalyst, 896
SiteCommand, 897
tabel de comparare, 895
Visual Site Map, 897
WebAnalyzer, 897
WebCollection, 897
WebCutter, 897
clic pe, 153
formulare, 205
hărți imagine client-side, 153
amestecarea cu hărți ce țin de server, 186
coordonate, 181, 184
creare, 180-185, 486
forme, 181-183
hiperlegături, 181
hărți imagine server-side, 152-153
amestecarea cu hărți ce țin de client, 186
creare, 185, 486
forme, 185
handlere, 186
standarde, 185
hiperlegături
cadre, 184
deactivating, 184
instrumente, 186-191, 194
instrumente de mapare la sit, 895
manipulare de la tastatură, 184
meniuri, 177
navigarea prin paginile Web, 486
partea clientului sau partea serverului, 180
problema accesibilității, 654-655
selectare conținut, 177-180
standarde (standarde NCSA), 185
timp de încărcare, mărire, 184
hărți imagine, 176
amestecarea hărților ce țin de server cu hărțile ce țin de client, 186
browsere non-grafice, 184
clic pe, 153
client-side (partea clientului), 153, 180-185, 486
coordonate, 181, 184
forme, 181-183, 185
formulare, 205
handlere, 186
hiperlegături, 181, 184
instrumente, 186-191, 194
keyboard manipulation, 184
Lynx, 741, 771
meniuri, 177
navigare (pe pagini Web), 486
problema accesibilității, 654-655
selectare conținut, 177-180
server-side (partea serverului), 152-153, 180, 185, 486
standarde, 185
viteză de descărcare, mărire, 184
hărți imagine cerc
partea clientului, 181
partea serverului, 185
hărți imagine NCSA
handlerul imagemap pentru harta imagine, 186
standarde, 185
hărți imagine poligon, 182
partea clientului, 181
partea serverului, 185
hărți imagine rectangulare, 182
partea clientului, 181
partea serverului, 185
hearing-impaired users and accesibility issues, 643-644
Help (evenimentul onhelp), 409
hh (format oră), 94
hidden elements (formulare), 202
hieroglife, 626
hiperlegături, 167
adrese e-mail, 95
ancore, 169-170
ancora destinație, 168

ancora sursă, 168
ancore denumite, 170-171
animație, 486-487
cadre, 169-170
controlarea acțiunii, 73
culori
active hyperlinks, 502
legături vizitate, 502
static hyperlinks, 502
documente XML față de documente HTML, 585-587
fișiere Adobe Acrobat, 173-174
fișiere audio, 173-174
fișiere tip imagine, 169, 172-174
hărți imagine, 486
cadre, 184
dezactivare, 184
partea clientului, 181
imagini, 484
întrebuințări, 167-168
legături tip text, 484
marcaje de listă, 484
noduri, 23
problema accesibilității, 656
structură, 168-169
hiperspațiu, 21
Home Page 3.0, 890
HomeSite, 890
Hot Dog Express, 889
Hot Dog Pro, 890
hotlinks, *vezi* hiperlegături
HoTMetal Pro, 889
HotSauce, 897
HTML (HyperText Markup Language)
caracteristici uzuale, 735
cele mai noi surse de informare, 800
documente, *vezi* documente HTML
Dynamic HTML, 325
editori, 887
Amaya, 890
BBEdit, 890
Composer, 889
Emacs, 893
funcționalitate, 236
Home Page, 890
HomeSite, 890
Hot Dog Express, 889
Hot Dog Pro, 890
HoTMetaL Pro, 890
Multi-Edit, 891, 893
PageMill, 890
selectare, 233
tabel de comparare, 887, 889
Vi, 893
Visual SlickEdit, 891
Webber, 35, 891
WebEdit Pro, 891
evoluție, 24
HTML +, 25
HTML 2.0, 26
HTML 3.0, 26-27
HTML 3.2, 27
HTML 4.0, 27
Level 0 HTML, 24
Level 1 HTML, 24-25
filozofie, 671
funcționalitate, 670
HTML 4.0 Specification W3C Working Draft, 670
internaționalizare, 624
aliniere, 638
atribute, 733
despărțirea, 638
direcția de scriere, 636-637
fonturi, 639-641
ghilimele, 637
identificarea limbii, 635-637
marcarea limbii, 624
MIME, 628-629
modul de scriere cursiv, 637-638
RFC 2070, 624
setul de caractere, 629-639
standarde de codificare a caracterelor, 624-628
limitări, adresare, 588-589
liste poștale, 350
Microsoft, 39
Netscape, 39
origini, 21, 22, 23
practică, 671
resurse online, 935
Jan's Guide to HTML, 937-938
Learning HTML by Example, 939
lista poștală, 939
scurtă prezentare a limbajului HTML, 937-938
situl As We May Think, 102
situl HTML Station, 937
situl Index DOT HTML, 937
situl Web Tools.org, 937
situl Xanadu, 936
tabel de comparare, 935-936
scripturi, folosirea etichetelor HTML, 417-418
standarde, 27
W3C, 27-28
validatori, 915-916
CAST Bobby 2.0, 917
CSE 3310, 916
tabel de comparare, 916
Web Techs, 916
Weblint, 917
variații, 39
XML, 582
avantaje HTML, 582
documente, 583
foi de stil, 587-588
legare, 585-587
obiectele de curs HTML/CSS, 604
validare, 583-585
HTML 4.0 Specification W3C Working Draft, 670
HTML Version entity (HTML 3.2), 40
HTML Writer's Guild, 933
HTML+, 25
HTTP(HyperText Transport Protocol), 22, 161-162
anteturi, 354
câmpuri de cerere, 354-355
câmpuri de răspuns, 355-356
cereri pentru aplicații CGI, 356
conexiuni, 354
limite, 414
origini, 22
standarde, 27
URL-uri, 161-162
HyperCard, 21-22
hyperfile, 21
HyperTalk, 21, 24
HyperText Markup Language, *vezi* HTML
Hypertext Transfer Protocol, *vezi* HTTP
hyperworld, 21
HyTime, 572

I

I-Pad eSoft, 927
IBM
cititoare de ecrane, 643-644
GML (limbajul generalizat de marcare), 36
identificare
declarații SGML, 37-38
sit Web, 471
identificator de fragment (legături XML), 577
identificatori extinși, 635-637
identificatori publici, 40-41
identificatori universali de resurse, 156
ideograme, 626-627
IDs (foile de stil JavaScript), 288
IEF component-based application, 386
ierarhia de instanțe (obiecte), 390
ierarhia obiectelor, 390
ieșire pentru sintetizatoare de vorbire, 74
ieșirea Braille, 74, 642-643
ieșirea ecran, 74
ieșirea print, 74
Illustrator, 900
Imaging Machine, 903
imagini, 147
aliniere, 50-52, 148-149, 514
design, 522-526
imagini încorporate, 525-526
la text, 523
orizontal, 523
vertical, 522
Barry's Clip Art Server, 905
explicitarea imaginilor, 653
forumul DisAbilities, 556-557
fundaluri, 820-821
fotografii (stoc de fotografii), 508
inserare, 213-215
granițe, 151, 514
hiperlegături
creare pornind de la imagini către fișiere cu imagini, 484
to image files, 169, 172-174
imagini GIF, 148
inserare, 213-215, 513-514
instrumente grafice, 506-507
Corel PhotoPaint, 507
DeBabelizer Pro, 507
LView Pro, 507
Paint Shop Pro, 506
Photoshop, 506
înălțime, 514
JavaScript, 324
layere, 302
lățime, 514
materialele sursă, 508
mărimi, 150-151, 504
obiecte, 153-154
problema accesibilității, 651
explicitări, 653-654
text descriptiv, 652-654
producere, etape, 508
calitatea, 508-509
dimensionarea imaginilor, 510-511
efecte speciale, 510
salvarea fișierelor imagine, 513
scanare, 508-509
randare progresivă, 513
resurse on-line
Homegurrr! page, 904
Icons by Goff page, 904
Photoshop Tips page, 904
software pentru grafică, 899-900
DeBabelizer, 901
Design Lab, 904
Fractal Design Painter, 901
GIF Construction Set, 902
GIFTool, 903
GIFTrans, 903
GIMP, 902
Illustrator, 900
Imaging Machine, 903
LView Pro, 903
Paint Shop Pro, 902
PHOTO-PAINT, 901
Photoshop, 900
tabel de comparare, 899-900
Web Wonk, 904
Web3, 902
XV Image BViewer, 903
spațiu, 151-152
spațiul orizontal, 514
spațiul vertical, 514
substituting for high-end content, 154-155
suportul browserului, 148, 394
text, plutirea în jur, 85-86
transparența, 513
viteza de descărcare, mărire, 59, 61
imagini cu note explicative, 653-654
imagini GIF (Graphics Interchange Format), 148
animație
creare, 219
inserare, 219-220

imagini GIF întrețesute, 513
salvare, 513
imagini GIF întrețesute, 513
imagini încorporate, 525-526
imagini PNG (Portable Network Graphics), 148
importul unei foi de stil XSL, 616
indentare, 604
indentări (text), 241-242
indexare
Gopher, 24-25
indexare asociativă, 20
cadre, 286
crearea cu coloanele tabelului, 136-138
indexarea asociativă, 20
indicatori
indicator de apariții, 43-44
indicator de minimalizare, 42
indicatori de apariții, 43-44
indicatori de minimizare, 42
indicatorul de apariții semnul de întrebare, 43-44
informație
informație lineară, 19
informație neliniară, 19
exemple, 479
sisteme de stocare, 20
World Wide Web, 22
informație secționată, 24
noduri, 23
informație liniară, 19
informație neliniară, 19
exemple de, 479
sisteme de stocare, 20
World Wide Web, 22
informație secționată, 24
informație tehnică, machetare, 53-54
inițiativa de codare a textului (TEI), 572
inline event handlers, 401, 409-412
inserare
animație, 213-215, 219-220
aplicații, 216
audio, 220
cod JavaScript, 327-330
cod VBScript, 337-338
controale ActiveX, 213, 217
fișiere PNG, 215
fișierul film QuickTime, 214
imagini, 213-215, 513-514
miniaplicații Java, 213, 216-217
plug-in, 213, 218-219
realitate virtuală, 213
scripturi de tratare a evenimentelor în documente HTML, 416-418
text
data și timpul modificării, 16, 94
marcarea textului inserat (editarea documentelor), 93
video, 214, 220
Institutul American de Standarde, 43-44, 876-877
instrucțiunea ATTLIST (DTDs), 45
instrumente
hărți imagine, 186-191, 194
instrumente grafice, 506
Corel PhotoPaint, 507
DeBabelizer Pro, 507
LView Pro, 507
Paint Shop Pro, 506
Photoshop, 506
JavaScript, 327
Web Embedding Font Tool, 535
instrumente grafice, 506-507
DeBabelizer, 507
LView Pro, 507
Paint Shop Pro, 506
PHOTO-PAINT, 507
Photoshop, 506
instrumente pentru maparea siturilor, 895
CLEARWeb, 896
Exploder Site Map, 897
Governor Site Map, 897
HotSauce, 897
Joe Barta's Site Map, 897
MAPA Service, 896
MapXsite, 896
SiteAnalyst, 896
SiteCommand, 897
tabel de comparare, 895
Visual Site Map, 897
WebAnalyzer, 897
WebCollection, 897
WebCutter, 897
instrumentul Live Image pentru crearea hărților imagine, 194
instrumentul Map Edit pentru crearea hărților imagine, 194
instrumentul MapMaker32 pentru crearea hărților imagine, 194
intefețe cu utilizatorul, *vezi* **câmpul interfaces**
interfețe, 460
analiza scopului, 462-467
anteturi, 470
componente, 466
formulare de răspuns, 469, 473
multimedia, 470
pagină de pornire, 467-468, 473
paginile de conținut, 468-469
contact, informații, 471
culori, 498
pagini stil în cascadă(CSS), 502
paletă de culori neprotejată, 504
paletă de culori protejate, 499-502
palete, 498
palete personalizate, 505-507
valori hexazecimale, 505-507
design
dimensiunea ecranului, 480
forme, 530-532
importanța, 479
motoare de căutare, 665-668

neliniaritate, 483-484
problema platformelor multiple, 480
proiectare pe orizontală, 481
proiectare pe verticală, 482-483
setările rezoluției, 480
spațiu alb, 473, 518-520
spațiu de lucru, 516-518
tipografia, 532-537
forumul DisAbilities, 539-541
identificarea sitului, 471
implementarea proiectului, 466
machetare, 479-480
motoare de căutare, 662
căutări prelucrate, 663
clasificarea rezultatelor căutării, 664
listele rezultatelor, 664-665
opțiuni de căutare, 663
navigare, 471
animație, 486-487
folosirea marginilor din dreapta pentru navigare, 489-491
folosirea marginilor din stânga pentru navigare, 489-491
folosirea metodelor de navigare multiple, 492, 495-497
folosirea părții de jos a paginilor pentru navigare, 488-489
folosirea părții superioare a paginilor pentru navigare, 488-489
hărți imagine, 486
identificarea zonei, 486
image links, 484
importanța, 479
legături tip marcaje de listă, 484
legături tip text, 484
machetare, 480
principiile proiectării, 460
claritate, 461, 472, 475
consecvență, 461, 472, 475
metaforă, 460, 472, 473
navigare, 462, 472, 475
orientarea, 462, 472, 475
scrierea sfaturilor, 471
scopuri, 460
interliniere, (text), 241, 537
Internet
Internet cu un browser Accent, 670
Internet Explorer, 671, 883-884
agitarea evenimentelor, 735
cele mai noi informații, 800
data binding capabilities, 437
degradare elegantă (DHTML), 386-387
eticheta <LINK>, 74
Model) (DOM), 380, 385-388, 735
Modelul de Obiect Document (Document Object
subrutine de tratare a evenimentelor, 409
URL-urile mailto, 163, 195
definiția, 464
situl Start Reading, 488
interogări (baze de date), 436
intrare
formulare, 196
JavaScript, 319
vezi și evenimente
IPO (Birou de Program Internațional), 381
ISO (International Standards Operation), 36
ISO 10646
formatul UTF-8, 633
internaționalizarea limbajului HTML, 627-628
ISO 639, 635
ISO 646
caracterele de control, 624-625
internaționalizarea limbajului HTML, 624-625
ISO 8879, 36
seria de standarde ISO 8859, 625
italic (text), 115, 223

Î

împachetarea cuvintelor celule (tabele), 124
înălțime
cadre, 277, 282-283
imagini, 59, 150-151, 514
linii de separare orizontale, 96
înglobare
documente HTML cu VBScript, 335-338
fișiere audio, 174
fonturi, 535, 639-640
scripturi JavaScript în documente HTML, 316-318
înregistrări (formulare de date), 438
editare, 452
filtrare, 438
vizualizare, 452
întreruperi de rând
creare, 85-87
liste, 102

J

Java, 316-317, 910-912
CGI (Common Gateway Interface), 353
JavaScript, integrare, 322-323
medii avansate, 487
miniaplicații
inserare, 213, 216-217
suportul browserului, 216
problema accesibilității, 658
programe păianjen, 661
resourse online, 912
Gamelan, 912

Haifa Research Institute, 912
Java Home Page, 912
Programmer's Source, 912
Servlet page, 912
Java Home Page, 912
JavaScript, 31, 288, 316, 332, 390, 836, 912-914
browsere
actualizarea proprietăților browserului, 320
detectarea caracteristicilor browserului, 320
suport, 318
capacități matematice, 322
casete de dialog, 320
cod, inserare, 327-330
comentarii, 292-295
declarații, 866
distincție între majuscule și litere mici, 287, 427
ECMAScript, 392
etichete HTML, 417
formulare și validarea datelor, 319
funcții, 865
ierarhia obiectelor, 391
imagini, 324
informații despre dată și oră, 322-323
instrumente, 327
înglobarea scripturilor în documentele HTML, 316-318
Java, integrare, 322-323
JavaScript World, 325
jocuri, 319
JScript, 318
metode (obiecte), 317
Microsoft Internet Explorer, 288
Modelul de Obiect Document (DOM), 392
Netscape, 288
obiectul anchor, 837
obiectul applet, 837
obiectul area, 837
elemente de tratare a evenimentelor, 837
metode, 837
proprietăți, 837
obiectul array, 838
metode, 838
proprietăți, 838
obiectul button, 838
obiectul checkbox, 839
obiectul combo
elemente de tratare a evenimentelor, 840
metode, 840
proprietăți, 839
obiectul date
metode, 840-842
proprietăți, 840
obiectul document
elemente de tratare a evenimentelor, 842
metode, 842
proprietăți, 841
obiectul FileUpload, 842
obiectul form, 843
obiectul frame
elemente de tratare a evenimentelor, 845
metode, 844-845
proprietăți, 844
obiectul function, 844
obiectul hidden, 845
obiectul history, 846
obiectul image
elemente de tratare a evenimentelor, 846-847
proprietăți, 846
obiectul layer
elemente de tratare a evenimentelor, 848
metode, 849
proprietăți, 846-847
obiectul link
elemente de tratare a evenimentelor, 849-850
proprietăți, 848
obiectul location
metode, 850
proprietăți, 850
obiectul math, 292
metode, 851
proprietăți, 850
obiectul mimeType, 852
obiectul navigator
metode, 853
proprietăți, 852
obiectul option, 853
obiectul password, 854
obiectul plug-in, 854
obiectul radio, 854
obiectul regular expression, 855
obiectul reset, 856
obiectul screen, 856-857
obiectul select, 857
obiectul string, 858-860
obiectul submit, 859-860
obiectul text
elemente de tratare a evenimentelor, 861
metode, 861
proprietăți, 859-860
obiectul textarea
elemente de tratare a evenimentelor, 861-862
metode, 861
proprietăți, 861
obiectul window
elemente de tratare a evenimentelor, 865
metode, 862-865
proprietăți, 861-862
operatori, 867
aritmetic, 1037
atribuire, 867
bitwise, 1037
logic, 868
precedență, 869
problema accesibilității, 658
programe păianjen, 661
proprietăți, 317
resurse online, 912-914
Hunting the SNARK, 914
JavaScript Debugger, 914
JavaScript Index, 914
JavaScript Planet, 913
JavaScript World, 914
Web Reference, 913
resurse
Cut-N-Paste JavaScript, 325
DevEdge Newsgroup FAQ, 327

Gamelan, 325
ghid de creație, 327
specificația limbajului JavaScript 1.1, 326
subrutine de tratare a evenimentelor, 317
user input, 319
versiunea 1.2, 318
versus alte limbaje de programare, 316-317
Visual JavaScript 1.0, 327
VRML, 322-323
jocul Mastermind, 319
jocuri
conștientizarea datelor, 435
JavaScript, 319
Mastermind, 319
Joe Bart's Site Map, 897
JScript, 318, 332, 782, 914-916
JScript Web Pages, 349
JSSS, *vezi* foile de stil JavaScript
Jughead, 24-25
justificare
linii de separare orizontală, 96
text, 68, 242-246, 521

K

K018-R codificarea caracterelor, 633
Kai's Power Tools, 510
kerning text, 537
Kirsanov, Dmitry, 502
Koenen, Mark, 491

L

Latin (ISO 8859-1) alfabet, 625
layere, 299
animație, 299
browsere, 299, 309
combinarea paginilor Web, 309-314
deplasare, 308-309
fundaluri, 302
imagini, 302
multiple, 303
ascundere, 305-307
ordinea de afișare, 305
poziționare, 300-301
prestabilit, 302-303
proiectare, 300
selectare, 314
layere opace, 302
lățime
cadre, 277, 282-283
celule (tabele), 127-128
coloane, 16, 134
imagini, 59, 150-151, 514
linii de separare orizontală, 96
tabele, 116
text (text monospațiat), 225
legarea datelor, 435
avantaje, 436
browsere, 437-438
conștientizarea datelor, 435
controlul pentru conștientizarea datelor Active X, 443
creare, 438-439
dezavantaje, 436
distincție între tipurile de date, 456-457
exemple, 439-440, 443-449
formulare de date, 438
actualizare, 546
câmpuri, 439
creare, 439
filtrare, 438
înregistrări, 438, 452
tabele dinamice
editare, 452
expansiune, 448-452
sortare, 453-456
legarea documentelor HTML, 73-74
legături
problema accesibilității, 656
XML
adresare, 575, 834
asociere cu informația, 574-575, 833
atributul href, 573
candidați, 576-577
comportare, 575, 834
documente interconectate, 578
domeniile pointerilor extinși, 577
grupurile de legături extinse, 835
identificatori de fragment, 577
legături extinse, 574, 598-599, 833
legături inline, 574, 834
legături simple, 573, 833
limbajul de legare XML, 572
pointeri extinși (indicatori extinși), 576, 834-835
recunoașterea legăturilor, 832-833
remaparea atributului, 833-834
resurse, 572
superlegături HTML, 585-587
vezi și hiperlegături
legături extinse (XML), 574, 598-599, 833
legături inline (XML), 574, 834
legături simple (XML), 573, 833
legende (form controls), 209-210, 656-657
Lemmon, Jerry, 76
letters (liste ordonate), 106
librăria Amazon.com, 350, 377
librăria Barnes & Noble, 350, 377
Limba arabă
alfabetul ISO 8859-6, 625
modul de scriere cursiv, 637
limbaj de specificare a semanticii și a stilului documentelor, 579
limbaje de marcare
definiția, 35

HTML
comparații cu XML, 582-588
limitări, adresare, 588-589
scopuri, 36
standarde, 36
XML, 582-588
limbaje de programare
CGI(Common Gateway Interface), 353
Java, 317
JavaScript, 316-317
VBScript, 333-334
Visual Basic, 333-334
limbaje de programare compilate, 316-318
limbaje de scriptare, 910
ECMAScript, 579
exemple, 617-619
folosește, 617-618
specificații, 617-618
JavaScript, 31, 288, 316, 332, 390
actualizarea proprietăților browserului, 320
browser support, 318
capacități matematice, 322
case-sensitivity, 427
casete de dialog, 320
cod, 327-330
comentarii, 318
detectarea caracteristicilor browserului, 320
ECMAScript, 392
etichete HTML, 417
ierarhia obiectelor, 391
imagini, 324
informații despre dată și oră, 322-323
instrumente, 327
integrarea cu Java, 322-323
jocuri, 319
JScript, 318
metode (obiecte), 317
Microsoft Internet Explorer, 288
Modelul de Obiect Document (DOM), 392
Netscape, 288
problema accesibilității, 658
proprietăți (obiecte), 317
resurse, 325-327
subrutine de tratare a evenimentelor, 317
user input, 319
validarea formularelor, 319
versiunea 1.2, 318
Visual JavaScript 1.0, 327
VRML, 322-323
vs alte limbaje de programare, 316-317
JScript, 332, 782
Livescript, 316, 332
scripturi de tratare a evenimentelor, 415
TCL, 418
VBScript, 317, 332-333
ActiveX, 347-348
case sensitivity, 427
cod, 327-330
documente HTML, înglobare, 335-338
etichete HTML, 417
funcții Split, 334
future, 392
mesajele din bara de stare, 343-345
Modelul de Obiect Document (DOM), 392
Modelul de Obiect pentru Scriptare Internet Explorer, 338-340
plug-in ScriptActive (Netscape), 334
resurse, 348-350
rularea scripturilor, 338-339
securitate, 334
suportul browserului, 332-335
validarea formularului, 340-343
variante, 333
viruși, 334
Visual Basic, 333-334
limbaje orientate spre obiecte, 317
limbajul de legare XML (XLL), 572
Limbajul de marcare chimic (CML), 580
Limbajul de marcare cu dispozitiv (HDML), 580
Limbajul de marcare matematic (MathML), 580
limbajul de programare TCL
CGI (Common Gateway Interface), 353
etichete HTML, 418
limbajul generalizat de marcare (GML), 36
limbi (limbi străine)
internaționalizarea limbajului HTML, 635-637
motoare de căutare, 662
limbi străine
browsere
browserul Tango, 670
Internet cu un browser Accent, 670
internaționalizarea limbajului HTML, 635-637
motoare de căutare, 662
line rules (tables), 117
linii (linii de separare orizontală), 96
linii de separare orizontală, 96
liniuța de dialog (-), 876-877
link popularity (motoare de căutare), 664
Lista Common Gateway Interface Listserv, 377
Lista de discuții Advanced CGI (Advanced CGI Discussion List), 377, 934
Lista de discuții Advanced HTML (Advanced HTML Discussion List), 377, 939

liste, 99
creare, 63, 66
listarea directoarelor, 113
liste glosar, 111-113
liste neordonate, 99
atribute, 108
creare, 108-109
marcaje de listă, 108-109
liste ordonate, 99
atribute, 100
creare, 99-102
domeniile numărului, 106
găzduire, 106-107
întreruperi de rând, 102
renumerotare, 104
sisteme, 106
text explicativ, 103
menu, 113
problema accesibilității, 647-648
rezultatele motoarelor de căutare, 664-665
liste de definire, *vezi* listele glosar
liste de selecție, *vezi* casetele listă
liste glosar, 111-112
liste neordonate, 99
atribute, 108
creare, 108-109
găzduire, 109
marcaje de listă, 108-109
liste numerotate, *vezi* liste ordonate
liste ordonate, 99
atribute, 100
creare, 99-102
domeniile numărului, 106
găzduire, 106-107
renumerotare, 104
sisteme, 106
text explicativ, 103
liste poștale
ActiveX, 350, 934
Advanced CGI Discussion List, 934
Advanced HTML Discussion list, 939
CGI, 377
CGI-List, 934
HTML, 350
VBScript, 348-350
VBScript, 350, 934
listinguri
cadre, comunicarea între cadre, 279
CGI (Common Gateway Interface)
creare baze de date, 370-375
numărător de acces, 362
numărător de acces, fișier HTML, 363
numărător de acces, script Perl, 362
rutine de tratare a formularelor, datele utilizatorului, 369
rutine de tratare a formularelor, eliminarea comenzii server-side include din datele utilizatorului, 370
conținutul paginilor Web, 59-60
copierea șabloanelor de pagini Web, 61-62
DHTML (Dynamic HTML)
animație în momentul încărcării, 418-420
interfețe cu utilizatorul, 430-431
prelucrare de formular, 422
scriptul de acțiune a utilizatorului, click and replace features, 428
scriptul de acțiune a utilizatorului, expand and collapse features, 424-426
dynamic table expansion, 450
dynamic tables, sorting, 453-456
editare tabele, 453
eticheta <TITLE>, 71
etichetele <!DOCTYPE> și <HTML>, 70
etichetele <HEAD> și </HEAD>, 70-71
evenimente, tratare de eveniment inline, 409
foi de stil în cascadă (CSS)
elemente suprapuse, 268
poziționarea relativă a elementului, 266
foi de stil, combinarea CSS cu JSSS, 296-298
foile de stil JavaScript, 293-294
caracteristici, 294-295
pagini Web, 291
proprietatea apply, 289-291
formatarea textului, 83
întreruperi de rând, 85
paragrafe, 84
text preformatat, 87-88
interfețe, structura HTML, 472-473, 476-477
JavaScript, 327-330
layere
combinarea paginilor Web, 309-314
deplasare, 308
fără elemente de control, 299
multiple, ascundere, 305-306
multiple, deplasare, 304
multiple, rearanjarea ordinii de afișare, 305
opac, 302-303
poziționare, 300-301
legarea datelor
o prezentare de înregistrări dinamică, de tipul indică și execută clic, 445-446
prezentarea înregistrării, 440-442
surse de date, 443, 447
liste, 63, 66
liste neordonate, 108-109
liste ordonate, 99
articole, 101
cu text explicativ, 103-104
găzduire, 107

întreruperi de rând, 102
renumerotare, 104-105
pagina Web cu informații tehnice, 53-54
pagină Web clasică, 52-53
pagină Web convențională, 49
pagină Web formală, 55-56
pagină Web modernă, 50-52
panglici, 57-58
structura paginii Web, 56
șabloane pentru pagini Web, 60-61
tabele
adăugarea conținutului, 134-136
adăugarea conținutului la tabelul index-plus-două-coloane, 139-140
setarea alipirii celulelor, 120
setarea alipirii coloanelor, 120-121
setarea alipirii rândurilor, 122
with newspaper-style columns, 144-145
VBScript
mesajele din bara de stare, 343-344
script pentru validarea datelor formularului, 342
subrutine în document HTML, 335-336
validarea datelor formularului, 340-343
viewing records, 452
litere, 871
LiveScript, 316, 332
looping video, 220
Lotus Notes/Domino, 920
lungimi (atributul font-size), 820
LView Pro, 507, 903
Lycos, 350, 375-377, 943

M

machetarea
cadre, 284, 529
pagini Web, 48-49, 55-56, 479
alinierea imaginii, 50-52
coloane, 52-53
informație tehnică, 53-54
liste, 63, 66
navigație, 480
panglici, 57-58
restricții, 480
spațiu, 49
șabloane, 56-57
text, 49-50, 59-60
problema accesibilității, 648
atributul align, 650-651
eticheta <BLOCKQUOTE>, 650
eticheta <PRE>, 650
tabele, 648-650
tabele, 528-529
MacHTTPD, 918-919
Macintosh GIFBuilder, 220
Macintosh nu este o mașină de scris, 537
macroinstrucțiuni (obiecte de curs), 613-614
Macromedia, 549, 907
Flash, 549
Freehand Graphics Studio, 902
plug-in Shockwave, 218-219
MacUser, 504
MacWorld, 908
Magellan, 943
Map This!, 186-193
MAPA service, 896
MapXsite, 896
marcaje de listă
liste, 108-109
superlegături, 484
marcare, definiția, 35
marcarea logică, 644-645
marcarea textului inserat, 224
margini
design, 518-520
foi de stil în cascadă CSS), 241-243
folosirea paginilor Web pentru navigare
cadre, 491
tabele, 489-491
formatarea, 94-96
îmbinarea cu partea de sus a browserului, 488
spațiu liber, 743
matematică (capacități JavaScript), 322
materialele sursă (graphic design), 508
mărime
celule (tabele), setare, 120
fonturi, 224, 226-228, 239-240, 535-537
imagini, 150-151, 504, 510-511
mărirea vitezei de încărcare a imaginilor, 59, 61
MCF (Meta Content Framework), 668-669
mecanismul de interogare, 22
medii avansate, navigarea în pagini Web, 486-487
medii, avansate, 486-487
meniu care se desfășoară (formulare), 205-207
meniuri
creare, 113
hărți imagine, 177
meniu care se desfășoară (formulare), 205-207
mesajele din bara de stare (VBScript), 343-345
Meta Content Framework (MCF), 668-669
metafora, principiu de proiectare, 460, 472, 474, 541
metalimbaje, 564
metoda abs( ), 851
metoda acos( ), 851
metoda addItem( ), 840
metoda alert( ), 844
metoda anchor( ), 858
metoda asin( ), 851
metoda atan( ), 851
metoda back( ), 846, 862
metoda big( ), 858
metoda blink( ), 858

metoda blur( ), 844
metoda bold( ), 858
metoda captureEvents( ), 842, 849, 862
metoda ceil( ), 851
metoda charAt( ), 858
metoda charCodeAt( ), 858
metoda Clear, 875
metoda clearInterval( ), 844, 862
metoda clearTimeout( ), 845, 865
metoda click( ), 838
metoda compile, 855
metoda concart( ), 838, 858
metoda confirm( ), 844, 862
metoda contextual (foile de stil JavaScript), 289
metoda cos( ), 851
metoda disableExternalCapture( ), 862
metoda enableExternalCapture( ), 862
metoda exec( ), 856
metoda exp( ), 851
metoda find( ), 862
metoda fixed( ), 858
metoda floor( ), 851
metoda focus( ), 838
metoda fontSize( ), 858
metoda forward( ), 846, 862
metoda fromCharCode( ), 858
metoda GET, 161-162
CGI (Common Gateway Interface), 357
metoda getDate( ), 840
metoda getDay( ), 840
formulare, 195
metoda getHours( ), 840
internaționalizare, 633
metoda getMinutes( ), 840
metoda getMonth( ), 841
metoda getSeconds( ), 841
metoda getSelection, 837
metoda getTime( ), 841
metoda getTimezoneOffset( ), 841
metoda getYear( ), 841
metoda go( ), 846
metoda HEAD, 161-162
metoda home( ), 862
metoda indexOf( ), 858
metoda italics( ), 858
metoda javaEnabled( ), 853
metoda join( ), 838
metoda lastIndexOf( ), 859
metoda link( ), 859
metoda load( ), 849
metoda log( ), 851
metoda match( ), 859
metoda max( ), 851
metoda min( ), 851
metoda moveAbove( ), 849
metoda moveBelow( ), 849
metoda moveBy( ), 849, 862
metoda moveTo( ), 849, 862
metoda moveToAbsolute( ), 849
metoda open( ), 844, 863-864
metoda parse( ), 841
metoda POST, 161-162
CGI(Common Gateway Interface), 357
formulare, 195
internaționalizare, 633
metoda pow( ), 851
metoda preference( ), 853
metoda print( ), 844, 864
metoda prompt( ), 844, 864
metoda Raise, 876
metoda random( ), 851
metoda releaseEvents( ), 843, 849, 864
metoda reload( ), 850
metoda removeItem( ), 840
metoda replace( ), 850, 859
metoda reset( ), 843
metoda resizeBy( ), 849, 864
metoda resizeTo( ), 849, 864
metoda reverse( ), 839
metoda round( ), 851
metoda routeEvent( ), 843, 849, 864
metoda scrollBy( )
metoda scrollTo( ), 864
metoda search, 859
metoda setDate( ), 841
metoda setHours( ), 841
metoda setInternal( ), 864
metoda setInterval( ), 844
metoda setMinutes( ), 841
metoda setMonth( ), 841
metoda setSeconds( ), 841
metoda setTime( ), 841
metoda setTimeout( ), 845, 865
metoda setYear( ), 841
metoda sin( ), 851
metoda slice( ), 839, 859
metoda small( ), 859
metoda sort( ), 839
metoda split ( ), 859
metoda sqrt( ), 851
metoda stop ( ), 865
metoda strike( ), 859
metoda sub ( ), 859
metoda submit ( ), 843
metoda substring, 859
metoda sup( ), 859-860
metoda tan( ), 851
metoda test( ), 856
metoda toGMTString ( ), 841
metoda toLocaleString ( ), 841
metoda toLowerCase( ), 859-860
metoda toUpperCase( ), 859-860
metoda UTC( ), 841
metode, 161-162
abs( ), 851
acos( ), 851
addItem( ), 840
alert( ), 844
anchor( ), 858
asin( ), 851
atan( ), 851
back( ), 846, 862
big( ), 858
blink( ), 858
blur( ), 844
bold( ), 858
captureEvents( ), 843, 849, 862
ceil( ), 851
charArt( ), 858
charCodeArt( ), 858
Clear, 875
clearInterval( ), 844, 862

clearTimeout( ), 845, 865
click( ), 838
compile, 855
concart( ), 838, 858
confirm( ), 844, 862
cos( ), 851
disableExternalCapture( ), 862
enableExternalCapture( ), 862
exec( ), 856
exp( ), 851
find( ), 862
fixed( ), 858
floor( ), 851
focus( ), 838
fontsize( ), 858
forward( ), 846, 862
fromCharCode( ), 858
getDate( ), 840
getDay( ), 840
getHours( ), 840
getMinutes( ), 840
getMonth( ), 841
getSeconds( ), 841
getSelection( ), 837
getTime( ), 841
getTimezoneOffset( ), 841
getYear( ), 841
go( ), 846
home( ), 862
indexOf( ), 858
italics( ), 858
javaEnabled( ), 853
JavaScript, 317
join( ), 838
lastIndexOf( ), 859
link( ), 859
load( ), 849
log( ), 851
match( ), 859
max( ), 851
metoda contextual (foile de stil JavaScript), 289
metoda GET ( ), 161-162
CGI (Common Gateway Interface), 357
formulare, 195
internaționalizare, 633
metoda HEAD, 161-162
metoda POST, 162
CGI (Common Gateway Interface), 357
formulare, 195
internaționalizare, 633
min( ), 851
moveAbove( ), 849
moveBelow( ), 849
moveBy( ), 862
moveby( ), 849
moveTo( ), 849, 862
moveToAbsolute( ), 849
open( ), 844, 863-864
parse( ), 841
pow( ), 851
preference( ), 853
print( ), 844, 863-864
prompt( ), 844, 864
raise, 876
random( ), 851
releaseEvents( ), 843, 849, 864
reload( ), 850
removeItem( ), 840
replace( ), 850, 859
reset( ), 843
resizeBy( ), 849, 864
resizeTo( ), 849, 864
reverse( ), 838
round( ), 851
routeEvent( ), 843, 849, 864
scrollBy( ), 864
scrollTo( ), 864
search( ), 859
setDate( ), 841
setHours( ), 841
setInterval( ), 844, 864
setMinutes( ), 841
setMonth( ), 841
setSeconds( ), 841
setTime( ), 841
setTimeout( ), 845, 865
setYear( ), 841
sin( ), 851
slice( ), 838, 859
small( ), 859
sort( ), 838
split( ), 859
sqrt( ), 851
stike( ), 859
stop( ), 865
sub( ), 859
submit( ), 843
substr( ), 859
substring( ), 859
sup( ), 859-860
tan( ), 851
test( ), 856
toGMTString( ), 841
toLocaleString( ), 841
toLowerCase( ), 859-860
toUpperCase( ), 859-860
UTC( ), 841
MicroEdge Visual SlickEdit, 891
Microsoft
ActiveX, 347-348
controale, inserare, 213, 217
controlul pentru conștientizarea datelor, 443
medii avansate, 457
resurse, 348-350
VBScript, 347-348
DHTML, 379
degradare elegantă, 386-387
legarea datelor, 435
Modelul de Obiect Document (DOM), 385-388
standarde, 380, 385-388
Front Page, 895
HTML, 39
JScript, 332, 349, 782
Modelul de Obiect Component (COM), 348
Modelul de Obiect Document (DOM), 392-394
OLE, 347
Site Builder Network, 464-467
SiteAnalyst, 896
VBScript, 332, 349
Visual Basic, 317
W3C, 27-28
Web Embedding Font Tool, 535

Microsoft Proxy server, 928
Microstar Near & Far, 591
MIME (Multipurpose Internet Mail
miniaplicații (miniaplicații Java), 213, 216-217
MKS (Mortice Kern Systems) Toolkit, 924
mm (format minute), 94
MM (format month), 94
modalități alternative de afișare a informației, 79
modele de conținut, 567-568
ambiguitate, 592
grupuri, 43-44
specificații, 42-43
modelul de eveniment (Modelul de Obiect Document), 397-398
Modelul de Obiect Component (COM), 348
Modelul de Obiect Document (DOM), 390
Modelul de Obiect pentru Scriptare Internet
modul de scriere cursiv (text), 637-638
moduri (XLS), 615-616
Mortice Kern Systems (MKS) Toolkit, 924
motoare de căutare (mecanisme de căutare), 350, 375-377, 941
Alta Vista, 662-664, 941
aplicații CGI, 360-361
Aport, 665
clasificarea documentelor HTML, 75
design, situri Web, 665
cadre, 668
eticheta <META>, 667
strategii pentru cuvintele cheie, 665-667
directoare, 661
Excite, 663, 668, 941
HotBot, 662-664
Infoseek, 664, 941
interfețe, 662
Lexis-Nexis, 943
liste de rezultate, 664-665
Lycos, 943
Magellan, 943
Meta Content Framework (MCF), 668-669
Northern Light Search, 664
opțiuni de căutare
căutare după frază, 663
căutare după mediu, 663
căutări prelucrate, 663
căutări speciale, 663
clasificare, 664
operatori booleeni, 663
proximitate, 663
restricții, 663
păianjeni de indexare, 661
baze de date, 662
înregistrare, 662
limbi străine, 662
limitări, 661-662
păianjeni eșantionați, 662
perioadele de actualizare, 662
Rambler, 667
Reuters, 943
tabel de comparare, 941
traficul generat, 660
WebCrawler, 664, 667
motorul de căutare (mecanismul de căutare) Infoseek, 350, 375-377, 664, 942
motorul de căutare Alta Vista, 350, 375-377, 662, 942
căutări prelucrate, 663
dicționar de scriere pe litere, 663
motorul de căutare Aport, 665
motorul de căutare C|Net's Search.com, 350, 375-377
motorul de căutare HotBot, 375-377, 662-664
motorul de căutare Northern Light Search, 664
motorul de căutare Rambler, 667
motorul de căutare WebCrawler, 664, 667
Movie Cleaner Pro, 909
MSDN, 925
multimedia, 470
forumul DisAbilities, 548
plug-in Shockwave, 218-219
resurse online
3DSite, 909
Cross-Platform Page, 909
MacWorld, 908
PCWorld, 909
WWW Virtual Library, 909
software, 906
3D Studio Max, 907
GoldWave, 909
Macromedia, 907
Movie Cleaner Pro, 909
NoteWorthy Composer, 909
Personal AVI Editor, 909
ReadAudio, 909
RealAudio, 906
Sound Forge, 909
tabel de comparare, 907
WebShot, 909
Multipurpose Internet Mail Extensions, *vezi* MIME

N

navigare
design, 462, 470-472, 475
forumul DisAbilities, 543, 549-556
navigarea cu taste de acces, 211-212
navigarea tab, 211
pagini Web
animație, 486-487
folosirea marginilor din dreapta pentru navigare, 489-491
folosirea marginilor din stânga pentru navigare, 489-491
folosirea metodelor de navigare multiple, 492, 495-497
folosirea părții de jos a paginilor pentru navigare, 488-489

folosirea părții superioare a paginilor pentru navigare, 488-489
hărți imagine, 486
identificarea zonei, 486
image links, 484
importanța, 479
legături tip marcaje de listă, 484
legături tip text, 484
machetare, 480
problema accesibilității, 656-657
navigarea folosind taste de acces
formulare, 211-212
problema accesibilității, 656-657
navigarea pe Web, 205, 940-941
navigarea tab și formularele, 211
Navigator, 216, 671, 884-885
cele mai noi informații, 800
JScript, 782
suport ActiveX, 217
suport Java, 216
NCSA Mosaic, 800, 884-885
Near & Far (Microstar), 591
Nelson, Ted, 21
NetObjects Fusion, 893-894
Netscape
DevEdge, 932
DHTML, 379
Modelul de Obiect Document (DOM), 383-386
standarde, 379, 383-386
eticheta <EMBED>, 213
eticheta <LINK>, 74
foile de stil în cascadă, 287
foile de stil JavaScript, 287
HTML, 39
JavaScript, 288, 316, 332
JavaScript debugger, 914
LiveScript, 316, 332
Meta Content Framework (MCF), 668-669
Modelul de Obiect Document (DOM), 390-392
Persistent Cookies Documentation, 939
plug-in
inserare, 213, 218-219
plug-in Shockwave, 218-219
problemele utilizatorului, 218-219
ScriptActive (VBScript), 334
VBScript, 392
Proxy server, 928
SuiteSpot servers, 919-920
W3C, 27-28
Netscape Communicator, 884-885
Netscape Composer, 890
Netscape Navigator, 216, 671, 884-885
cele mai noi informații, 800
JScript, 782
suport ActiveX, 217
suport Java, 216
Network News Transfer Protocol, *vezi* NNTP
NNTP (Network News Transfer Protocol), 163
noduri, 23-24
nonlinearitate în paginile Web, design, 483-484
notații în declarații XML, 832-833
note, creare, 138-139
NoteWorthy Composer, 909
Novell Border Manager, 927
NSCA HTTPD, 918-919
numărătoare (aplicații CGI), 361-363
numărătoare de acces în aplicații CGI, 361-363
nume universale de resurse, 156
numele stilurilor (XSL), 615
numere, formatare în tabele, 118
numire
ancore (hiperlegături), 170-171
tabele, 124

O

obiecte, 390
ancoră, 837
buton
elemente de tratare a evenimentelor, 839
metode, 839
proprietăți, 838
cadru
elemente de tratare a evenimentelor, 845
metode, 844-845
proprietăți, 844
casetă de dialog
elemente de tratare a evenimentelor, 839
metode, 839
proprietăți, 839
combo
elemente de tratare a evenimentelor, 840
metode, 840
proprietăți, 839
date
metode, 840
proprietăți, 840
document, 394
elemente de tratare a evenimentelor, 843
foile de stil Java Script (JSSS), 391
metode, 843
proprietăți, 841
elemente de tratare a evenimentelor (JavaScript), 317
Err, 876
expresie obișnuită
indicatoare, 855
metode, 855
parametri, 855
proprietăți, 855
FileUpload, 843
foile de stil JavaScript (JSSS), 288
formular, 843
funcție, 845
hidden, 845
ierarhie, 390

imagine, 153-154
elemente de tratare a evenimentelor, 846-847
proprietăți, 846
istoric, 394, 846
layer
elemente de tratare a evenimentelor, 849-850
evenimente, 848
legătură
metode, 849
proprietăți, 847-848
locație, 850
matematică
JavaScript, 292
metode, 851
proprietăți, 850
metode (JavaScript), 277
miniaplicație, 837
Modelul de Obiect pentru Scriptare Internet Explorer, 338-340
navigator
metode, 853
proprietăți, 850
obiectul all, 392
option, 853
password, 854
plug-in, 854
proprietăți (JavaScript), 317
radio, 854
reset, 856
screen, 856-857
select, 857
string, 858-860
submit, 859-860
suprafață, 837
elemente de tratare a evenimentelor, 837
metode, 837
proprietăți, 837
șir, 838
metode, 838
proprietăți, 838
text
elemente de tratare a evenimentelor, 861
metode, 861
proprietăți, 859-860
textarea
elemente de tratare a evenimentelor, 861-862
metode, 861
proprietăți, 861
tipul MIME, 852
window
elemente de tratare a evenimentelor, 865
metode, 862-865
proprietăți, 861-862
obiecte de curs
macroinstrucțiuni, 613-614
XSL, 601
caracteristici, 604
obiectele de curs DSSSL, 602-604
obiectele de curs HTML/CSS, 604
obiecte document (foile de stil JavaScript), 391
obiectul all, 392
obiectul applet, 837
obiectul area, 837
elemente de tratare a evenimentelor, 837
metode, 837
proprietăți, 837
obiectul button
elemente de tratare a evenimentelor, 838
metode, 838
proprietăți, 838
obiectul checkbox
elemente de tratare a evenimentelor, 839
metode, 839
proprietăți, 839
obiectul combo
elemente de tratare a evenimentelor, 840
metode, 840
proprietăți, 839
obiectul date
metode, 840-842
proprietăți, 840
obiectul document, 394
elemente de tratare a evenimentelor, 842
metode, 842
proprietăți, 841
obiectul Err, 876
obiectul form
elemente de tratare a evenimentelor, 843
metode, 843
proprietăți, 843
obiectul frame
elemente de tratare a evenimentelor, 845
metode, 844-845
proprietăți, 844
obiectul history, 394
metode, 846
proprietăți, 846
obiectul image
elemente de tratare a evenimentelor, 846-847
proprietăți, 846
obiectul link
elemente de tratare a evenimentelor, 849-850
proprietăți, 848
obiectul location
obiectul Location (Modelul de Obiect pentru Scriptare Internet Explorer), 339
metode, 850
proprietăți, 850
obiectul math
JavaScript, 292
metode, 851
proprietăți, 850
obiectul mimeType, 852
obiectul navigator
obiectul Navigator (Modelul de Obiect pentru Scriptare Internet Explorer), 339
metode, 853
proprietăți, 852
obiectul option, 853
obiectul password, 854
obiectul plug-in, 854
obiectul radio, 854
obiectul regular expression
indicatoare, 855
metode, 855
parametri, 855
proprietăți, 855

obiectul reset, 856
obiectul screen, 856-857
obiectul select, 857
obiectul string, 858-860
obiectul submit
  elemente de tratare a evenimentelor, 859-860
  metode, 859-860
  proprietăți, 859-860
obiectul text
  elemente de tratare a evenimentelor, 861
  metode, 861
  proprietăți, 859-860
obiectul textarea
  elemente de tratare a evenimentelor, 861
  metode, 861
  proprietăți, 859-860
obiectul window
  elemente de tratare a evenimentelor, 865
  metode, 862, 864-865
  proprietăți, 861-862
objectul layer
  evenimente, 848
  metode, 849
  proprietăți, 847-848
OLE, 347
omitere
  eticheta <BODY>, 77-78
  eticheta <HEAD>, 77-78
  etichete
    etichete de început, 42
    etichete de încheiere, 42
    SGML, 38-39
OMITTAG YES, 38-39
Open Software Description (OSD), 580
OpenTag, 580
operator de împărțire (/), 876
operatori, 876-877
operatori bitwise, 1037
operatori booleeni (motoare de căutare), 663
operatori de atribuire, 867
operatori de precedență, 869
operatori logici, 868
  aritmetic, 1037
  atribuire, 867
  bitwise, 1037
  logic, 868
  precedență, 869
operatorul Eqv, 876-877
operatorul Imp, 876-877
operatorul Mod, 876-877
operatorul Sau, 876-877
operatorul și, 876
operatorul Xor, 876-877
Organizația internațională de standarde, *vezi* ISO
internaționalizarea limbajului HTML, 37-38, 624
  aliniere, 638
  atribute, 733
  despărțirea, 638
  direcția de scriere, 636-637
  fonturi, 639-641
  ghilimele, 637
  identificarea limbii, 635-637
  marcarea limbii, 624
  MIME, 628-629
  modul de scriere cursiv, 637-638
  RFC 2070, 624
  seturile de caractere
    formulare, 631-633
    probleme, 631-634
    setul de caractere externe, 629-632
    setul de documente al caracterului, 629-631
  standarde de codificare a caracterelor, 624
    codificări pe 16 biți, 626-627
    codificări pe 8 biți, 625
    codul ASCII pe 7 biți, 624-625
    combinații de biți, 624-625
    ISO 10646, 627-628
    setul de codificare a caracterelor, 624
ordering elements, 269
ordinea de afișare a layerelor, 305
ordinea de tabulare și probleme de accesibilitate, 656-657
orientare
  design, 462, 472, 475
  forumul DisAbilities, 543
OUTER SELECTOR, 816
Output media types, 74
O'Reilly WebSite 2.0, 918-919

P

padding
  atribute, 823
  celule (tabele), 522
PageMill, 890
pagina cu întrebări frecvente, 491
pagina de felicitare, *vezi* pagina de bun venit
pagina Web A Beginner's Guide HTML, 349, 374
Pagina Web Ask the VB Pro, 349
pagina Web Cnets ActiveX Resources Pages, 350
pagina Web NCompass Labs, 349
pagina Web Netscape's Persistent Cookies Documentation, 350
pagina Web Rollins & Associate's, 349
pagina Web Scribe - The VBScript Writers Resource, 349
pagina ZDNet's Active X Files, 350
pagină de întâmpinare, 467-468, 473
pagini Web
  anteturi, 470
  aplicații CGI (Common Gateway Interface), integrare, 354
  audiență, 249-250, 462-467
  autor, identificare, 77
  căutare, 205
  combinarea cu machetele, 309-314
  contact information, 471

copiere, 61-63
copyright, 77
creare, 59-60, 61
culori, 498
  foile de stil în cascadă (CSS), 502
  paletă de culori protejată, 499-501
  palete, 498
  valori hexazecimale, 505-507
descriere, 77
DHTML delivery, 414-415
foi de stil JavaScript, 290-296
formulare de răspuns, 469, 473
fundaluri, 491
HTTP delivery, 414
identificarea sitului, 471
încărcare, 404
machetare, 48-49, 479
  alinierea imaginilor, 50, 52
  coloane, 52-56
  informație tehnică, 53-54
  liste, 63, 66
  panglici, 57-58
  restricții, 480
  spațiu, 49
  text, 49-50, 59-60
multimedia, 470
navigare, 471
  animație, 486-487
  folosirea marginilor din dreapta pentru navigare, 489-491
  folosirea marginilor din stânga pentru navigare, 489-491
  folosirea metodelor de navigare multiple, 492, 495-497
  folosirea părții de jos a paginilor pentru navigare, 488-489
  folosirea părții superioare a paginilor pentru navigare, 488-489
  hartă imagine, 486
  identificarea zonei, 486
  image links, 484
  importanța, 479
  legături tip marcaje de listă, 484
  legături tip text, 484
  machetare, 480
pagină de întâmpinare, 467-468, 473
pagini Web dinamice
  acțiunile utilizatorului, 424-425
  creare, 415-418
  interfețele cu utilizatorul, 430-431
  legarea datelor, 438-439
  procesarea formularelor, 422-424
  timpul de încărcare a paginilor Web dinamice, 418, 422
paginile de conținut, proiectare, 468, 469
proiectare
  dimensiunea ecranului, 480
  forme, 530-532
  importanța, 479
  neliniaritate, 483-484
  problema platformelor multiple, 480
  proiectare pe orizontală, 481
  proiectarea pe verticală, 482-483
  setările rezoluției, 480
  spațiu alb, 518-522
  tipografia, 532-537, 537
  work space, 516-518
reîncărcarea automată, 76
șabloane, 56-57
titluri, 57-58, 71-72
titluri multiple, 72
unloading, 404
URL-uri, 161-162
writing, 471
writing tips, 72
vezi și interfețe
pagini Web dinamice
  acțiunile utilizatorului, 424-425
  creare, 415-418
  interfețe cu utilizatorul, 430-431
  legarea datelor, 438-439
  momentul încărcării paginilor Web dinamice, 418, 422
  prelucrare de formular, 422-424
pagini Web VBScript, 349
Paint Shop Pro, 506, 902
paleta de culori neprotejată, 504
paletă de culori protejată, 499-501, 556
palete, *vezi* culori, palete
panglici, creare, 57-58
parafocuri, 465, 926
  Check Point Firewall, 928
  Cisco Systems PIX, 928
  I-Pad e Soft, 927
  Novell Border Manager, 927
paragrafe, 42
  aliniere, 85
    aliniere la dreapta, 522
    aliniere la stânga, 522
    alinierea pe centru, 522
    design, 520
  creare, 84-85
parametrul BASESET (Declarațiile SGML), 40
paranteze unghiulare, 30
parent frames, 281
password entry boxes in forms, 203-204
path (eticheta <BASE>), 73
păianjeni de indexare, 661
  baze de date, 662
  hiding information from, 668
  înregistrarea, 662
  limbi străine, 662
  limitări, 661-662
  păianjeni eșantionați, 662
  perioadele de actualizare, 662
păianjeni eșantionați, 662
părinți (etichete HTML), 395-396
PC-ul nu este o mașină de scris, 537

PCWorld, 909
pe partea clientului
hărți imagine
amestecarea cu hărți ce țin de server, 186
creare, 180-185, 486
forme, 182-183
formulare, 205
instrumente, 186-191, 194
Lynx, 741, 771
navigare, 486
problema accesibilității, 654
versus server-side maps, 180
scripturi, apelarea aplicațiilor CGI, 359
pe partea serverului
aplicații (CGI), 359-360
hărți de imagine
amestecarea cu hărți ce țin de client, 186
creare, 185, 486
handlere, 186
instrumente, 186-191, 194
navigare, 486
standarde, 185
versus client-side, 180
server-side include (SSIs), 76, 359
Perl
CGI (Common Gateway Interface), 353
resurse, 375-377
resurse online, 923
MKS Toolkit, 924
Perl CGI Examples site, 924
Perl home page, 924
Perl Tutorial, 924
Perl/Tk FAQ, 924
Web Engineer's Toolbox, 924
scriptul BotWatch Perl, 662
PERL CGI, 162
Personal AVI Editor, 909
PHOTO-PAINT, 901
PhotoPaint (Corel), 507
PhotoShop 4.0, 900
ca instrument de desen, 506
filtre, 510
paletă de culori protejată, 498
palete personalizate, 502
tehnici de iluminare, 510
pixeli (rezoluție), 516
plasare, *vezi* poziționare
platforme
browsere, 671
Web page design, 480
plug-in ScriptActive (VBScript), 334
pointerii extinși (indicatori extinși) (legături XML), 576, 834-835
domenii, 577
locația relativă, 577, 835
locație absolută, 576
POP3 (Post Office Protocol 3), 628
Portable Network Graphic, *vezi* imagini PNG
porturi (servere), 161-162
Post Office Protocol 3 (POP3), 628
Power Tools (Kai's), 510
poziționare
elemente
elemente suprapuse, 268-269
foi de stil în cascadă (CSS), 261
ordering elements, 269
poziționare absolută 318-322
poziționare relativă, 261-262, 265-268
scripturi, 300-301
imagini in fundaluri, 821
layere, 300-301
navigare în pagini Web, 488
poziționarea absolută a elementului (foi de stil în cascadă), 261-262
poziționarea relativă a elementului (foi de stil în cascadă), 261-262, 265-268
prăjituri
JavaScript, 322
programe păianjen, 661
VBScript, 346-347
predefiniri
atribute în XML DTDs, 569-570
atribute XML, 830
atributele elementelor, 45
fonturi, 535
layere, 302-303
lățimile celulelor tabelului, 127-128
XSL, 614
prelucrare
documente XML, 828
conformitate, 832-833
tratarea entităților, 832-833
formulare, 194-195
XSL, 579-580
prelucrare de formular, 422
problema accesibilității
audio, 658
cadre, 657, 756-757
convenabil, 643-644
dispozitive, 642-643, 656-657
documente text integrale, 658-659
emoticonuri, 646
formulare, 656
<eticheta FIELDSET>, 656-657
<eticheta LABEL>, 656
<eticheta LEGEND>, 656-657
hărți imagine, 654, 656
imagini, 651, 652
explicitarea imaginilor, 653-654
text descriptiv, 652-654
Java, 658
JavaScript, 658
machetarea, 648
atributul align, 650-651
eticheta <BLOCKQUOTE>, 650
eticheta <PRE>, 650
tabele, 648-650

marcarea logică, 644-645
modalități alternative, 644-645
principiile unitare de accesibilitate a sitului Web, 645
punctuație, 646
superlegături, 656
tehnologii asistive, 643-644
text, 645
culori, 646-647
formatarea, 645
liste, 647-648
marcajul, 645
video, 658
vizibilitate, 642-643
procentaje
atributul font-size, 820
atributul vertical-align, 822
procesoare XML de nevalidare, 832-833
procesoare XML de validare, 832-833
profesioniștii în Web, 506
programe
baze de date, 662
înregistrare, 662
limbi străine, 662
păianjeni eșantionați, 662
perioadele de actualizare, 662
IEF component-based application, 386
vezi și aplicații
programe păianjen (motoare de căutare), 661
programul Web3, 902
proiectanți de grafică
tips for developing design skills, 506
tips for transferring design skills, 505-506
proiectare
anteturi, 470
componente, 466-470, 473
considerații despre motoarele de căutare, 665-668
cross-browser design, 480, 735-736
culori, 498
browser-based palettes, 504
custom palettes, 505-507
pagini stil în cascadă (CSS), 502
paletă de culori neprotejată, 504
paletă de culori protejată, 499-501
palete, 498
valori hexazecimale, 505-507
dimensiunea ecranului, 480
forme, 530-531
cercuri, 531
dreptunghiuri, 531
triunghiuri, 531
fotografii (stoc de fotografii), 508
graficienii profesioniști, 505-506
imagini
dimensiune, 504
production steps, 508-513
randarea progresivă, 513
transparență, 513
implementare, 466
importanța, 479
instrumente grafice, 506
Corel PhotoPaint, 507
DeBabelizer Pro, 507
LView Pro, 507
Paint Shop Pro, 506
Photoshop, 506
layer, 300
machetare, 479
cadre, 529
problema accesibilității, 648-650
tabele, 528-529
materialele sursă, 508
neliniaritate, 483-484
principii, 460-463, 472, 474-475
claritate, 461, 472, 475, 541
consecvență, 461, 472, 475, 542-543
metaforă, 460, 541
navigare, 462, 472, 475
orientare, 462, 472, 475
profesioniști în Web, 506
proiectare pe orizontală, 481
proiectarea pe verticală, 482-483
scop, 544
setările rezoluției, 480
spațiu, 49
spațiu alb, 473, 518
margini, 518-520
proporția, 526
proximitatea, 526
tipografia, 532-537, 537
proiectare pe orizontală (pagini Web), 481
proiectare pe verticală (pagini Web), 483
proiectul Gutenberg, 235
prologuri (documente XML), 828
proporția, design, 526
proprietatea apply (foile de stil JavaScript), 289-291
proprietatea Description (obiectul Err), 876
proprietatea Number (obiectul Err), 876
proprietatea Source (obiectul Err), 876
proprietatea z-index a stilului, 269
proprietăți
JavaScript, 317
obiectul anchor, 837
obiectul applet, 837
obiectul area, 837
obiectul array, 838
obiectul button, 839
obiectul checkbox, 839
obiectul combo, 839
obiectul document, 841
obiectul FileUpload, 843
obiectul form, 843
obiectul frame, 844
obiectul function, 845
obiectul hidden, 845

obiectul history, 846
obiectul image, 846
obiectul layer, 847-848
obiectul link, 848
obiectul location, 850
obiectul math, 850
obiectul mimeType, 852
obiectul navigator, 852
obiectul option, 853
obiectul password, 854
obiectul plug-in, 854
obiectul radio, 854
obiectul RegExp, 855
obiectul Regular Expression, 855
obiectul reset, 856
obiectul Screen, 856-857
obiectul select, 857
obiectul String, 858
obiectul submit, 859-860
obiectul text, 859-860
obiectul textarea, 861
obiectul window, 861-862
proprietatea apply (foile de stil JavaScript), 289-291
proprietatea z-index a stilului, 269
proprietățile obiectului FileUpload, 842
proprietățile obiectului function, 845
proprietățile obiectului hidden, 845
protocoale de transfer, *vezi* scheme
protocol (eticheta <BASE>), 73
protocol de control al transmisiei/protocol Internet, *vezi* **TCP/IP**
protocol Internet), 24-25
proximitate, design, 526
publicații online
Albuquerque Alibi, 492
archived issues, 479
MacUser, 504
Tucson Weekly, 492
Wired, 498
punctuație, probleme de accesibilitate, 646
push, butoane (formulare), 197-198, 205

R

Rabbit Internet's Web Site Explorer, 897
Raggett, Dave, 26
randarea progresivă, 513
ranking search results (motoare de căutare), 664
rânduri
cadre, 275
tabele, 116, 133
aliniere, 118
alipire peste, 122
creating, 118
formatare, 116
grupare, 124
realitate virtuală
inserare, 213
interpretarea browserului, 213
rearanjare layere, 305
redimensionarea cadrelor, 276-277
referințe
caractere în documente XML bine-formate, 832
entități în documente XML bine-formate, 832
SGML DTDs, 37
referiri la caractere, 41
reguli (tabele), 117
reguli de stil (HSL), 606, 614-615
moduri, 615-616
numele stilurilor, 615
regulile de construcție (XSL), 579, 605-613
acțiuni, 610-611
filtre, 612-614
predefiniri, 614
modele, 607-608
atribute, 608-609
calificative, 609-610
priorități, 610
reîncărcarea automată a paginilor Web, 76
relațiile dintre seturile de cadre, 280
relevancy boosters (motoare de căutare), 664
remapare
atribute în XML DTDs, 569
atribute XML, 833-834
renumbering ordered lists, 104
repetarea imaginilor în fundal, 821
restricții pentru motoarele de căutare, 663
resurse (legături XML), 572
imagini tip PNG, 148, 213
Rețele Intranet, definiția, 465
Reuters, 943
revista DATAMATION, 21
reviste (online)
archived issues, 479
MacUser, 504
Wired, 498
rezoluție
design, 516
pixeli, 516-518
scanarea imaginilor, 510
RFC 2070, 37-38, 624
caracteristicile folosirii mai multor limbi, 636
instrumente de marcare direcțională, 636-637
roboți, 661

S

salvarea fișierelor, 513
scanarea imaginilor, 508-510
schema URL callto, 166
schema URL televiziune, 166
schema URL TV, 165
scheme (URL-uri), 161
callto, 166
fișier, 164
FTP, 164
HTTP, 161-162
mailto, 162-163
news, 163
NNTP, 163
RealAudio, 173-174
telnet, 164
TV, 165
scriptul BotWatch al firmei Perl, 662
scripturi
definire, 79
elemente de poziționare (CSS), 270-272
etichete HTML, 417-418
JavaScript, înglobarea în documente HTML, 316-318
Perl (scriptul BotWatch Perl), 662
scripturi CGI pentru formulare de validare a datelor, 343
scripturi de partea clientului pentru apelarea aplicațiilor, 359
scripturi de tratare a evenimentelor, 415
animație în momentul încărcării, 418, 422
browser support, 432-433
comentarii, 433-434
inserting in HTML documents, 416-418
interfețe cu utilizatorul, 430-432
limbaje de scriptare, 415
prelucrare de formular, 422-424
scripturile de acțiune a utilizatorului, 424-425
VBScript
ascunderea scripturilor față de browsere incompatibile, 335-336
rulare, 338-339
scripturi CGI și validarea datelor formularului, 343
scripturi de tratare a evenimentelor, 415
animație în momentul încărcării, 418, 422
browser support, 432-433
comentarii, 433-434
inserare în documente HTML, 416-418
interfețe cu utilizatorul, 430-432
limbaje de scriptare, 415
prelucrare de formular 517-519
scripturile de acțiune a utilizatorului, 424-425
scripturile de acțiune a utilizatorului (DHTML), 424-425
scripturile de interfață cu utilizatorul (DHTML), 430-432
Search Engine Watch, 660
searchpart (procesarea CGI), 161-162
secțiunea CAPACITY (SGML), 37-38
secțiunea CHARSET (declarații SGML), 37-38
secțiunea FEATURES (SGML), 38-39
secțiunea SCOPE (SGML), 38-39
secțiunea SYNTAX (SGML), 38-39
secțiuni condiționale în DTD-uri XML, 564, 830-831
secțiuni cu declarații SGML
secțiunea CAPACITY, 37-38
secțiunea CHARSET, 37-38
secțiunea FEATURES, 38-39
secțiunea SCOPE, 38-39
secțiunea SYNTAX, 38-39
secțiunile CDATA
documentul XML, 828
XML DTDs, 572
Secure Sockets Layer, *vezi* **SSL**
securitate
parafocuri, 926
Check Point Firewall, 928
Cisco Systems PIX, 928
I-Pad eSoft, 927
Novell Border Manager, 927
SSL
Apache SSL, 928
SSLaps, 928
SSLeay, 928
VBScript, 334
WWW Security FAQ, 927
selectare
layere, 314
text, 408
selectori, 816
atribute, 816-817
clase, 816
etichete, 816
selectori contextuali, 816
valori, 816
selectori contextuali, 816
selectorul INNER, 816
Selena Sol's CGI Script situl Archive Web, 375-377, 923
self frames, 280
semantics of SGML, 37
semnul egal (=), 871, 876-877
semnul mai mare sau egal (>=), 876-877
semnul mai mic decât (<), 876-877
semnul minus (-), 876-877
semnul plus (+) indicator de apariție, 43-44, 876
separator T (format timp), 94
separatori
elemente, 43-44
text (linii de separare orizontală), 96
servere, 917
Apache, 919-920
cereri HTTP pentru aplicații CGI, 356
CERN HTTPD, 918
CL-HTTP, 919-920
Cold Fusion, 920
directive, 358
familia de servere Sun, 919-920
Jigsaw, 919-920
Lotus Notes/Domino, 920

MacHTTPD, 918-919
NSCA HTTPD, 918-919
O'Reilly WebSite 2.0, 918-919
porturi, 161-162
proxy, 928
servere de date, current delivery methods, 436
servere de pictograme, 639
SuiteSpot, 919-920
tabel de comparare, 918
tehnologii de sprijin, 418
WebStar, 918-919
servere proxy, 928
servere Web
cereri HTTP pentru aplicații CGI, 356
directive, 358
serverele SuiteSpot, 919-920
servername (eticheta <BASE>), 73
Serverul Apache, 919-920
Serverul Common Lisp Hypermedia (CL-HTTP), 919-920
setarea char (atributul align al <TABLE>), 118
setul de caractere
caractere non-Latine (HTML), 37-38
datele caracterului, 43
documente XML, 827
DTDs, 41
formulare, precizare, 196
internaționalizarea limbajului HTML
formulare, 631-633
probleme, 633-634
setul de caractere al documentului, 629-631
setul de caractere externe, 630, 629-632
înlocuibile, datele caracterului, 43
MIME, 628
SGML, 37-38
seturile de câmpuri (form controls), 209-210
SGML "ISO 8879 1986"
declarații SGML, 37-38
SGML (Standard Generalized Markup Language), 23, 35
analizoare, 38-39
citire, 38-39
comentarii, 37-38
declarații, 37
declarații, 37
identificare, 37-38
parametrul BASESET, 40
secțiunea CAPACITY, 37-38
secțiunea CHARSET, 37-38
secțiunea FEATURES, 38-39
secțiunea SCOPE, 38-39
secțiunea SYNTAX, 38-39
standard, 37-38
documente, 37, 40
DTDs, 24, 37
elemente, 37
etichete, 684, 737
funcționalitate, 37
GML (limbajul generalizat de marcare), 36
limitări HTML, 588
origini, 36
semantics, 37
sintaxă, 37
XML, 564
shareware, 897
C|Net Download.com, 898
TUCOWS Central, 898
ZDNet Software Library, 898
Shockwave, 213, 218-219, 487
SHORTTAG YES, 38-39
Simple Mail Transfer Protocol (SMTP), 628
sintaxa
foi de stil în cascadă (CSS)
definiții, 232, 238, 816
pagini stil inline, 817
pagini stil înglobate, 817
pagini stil legate, 817
SGML, 37
tabele, 132
XML, 827
conformitate, 832-833
documente, 828
legături, 833-834
structuri fizice, 831-832
structuri logice, 829-830
Sirius Multimedia Links, 909
sistemul de avizier electronic pentru forumul DisAbilities, 546-547
Site Builder Network (Microsoft), 464-467
SiteCommand, 897
SiteMill, 893-894
Situl Alaskan Flowers, 560
situl Andy's Netscape HTTP Cookie Notes, 350, 939
situl Arizona-Sonora Desert Museum, 469
Situl As We May Think, 936
situl Bitstream, 239, 639
situl Blindness Resource Center, 933
situl Boutell.com Archive Resources, 375-377
Situl C++ CGI Class Library, 375-377
situl Caboodles, 471
situl CGI Manual of Style Web, 375-377
situl cgi-lib.pl Perl CGI Library, 375-377
situl CGI.pm Perl 4 CGI Library, 375-377
situl Common Gateway Interface, 375-377
situl Compendium of HTML Elements, 937
situl Cross-Platform Page, 909
situl Cut-N-Paste JavaScript, 325, 914
situl DesktopPublishing.Com, 537
situl DevEdge, 932

situl DOM, 930
Internet Explorer, 735
istoric, 390
Microsoft, 392-394
Netscape, 390-392
Microsoft, 380, 385-388
Netscape, 383-386
standard status, 394
cerințe, 394-395
manipularea conținutului și a documentului, 396-397
modelul de eveniment, 397-398
structure navigation, 395-396
W3C, 379-383
situl ECMA, 617-618
situl Free Database Gateway List, 375-377
situl Free Icons by Goff, 904
situl Gameland, 325
situl Governor Map, 897
situl Grafica Obscure, 909
situl Graphics on the Web, 931
situl GSQL-A Mosaic-SQL Gateway, situl Guide to Publishing HTML and Forms, 937-938
situl Guide to Web Style, 937
situl Homegurrrl, 904
situl HTML Guru, 933
situl HTML Quick Reference, 349, 375-377, 937-938
situl HTML Station, 937
situl HTML Workshop and Survival Guide, 933
situl Hunting the SNARK, 914
situl Hypertext Markup Language, 349, 375-377
situl Image Club, 239, 508
situl Index DOT HTML, 937
situl Index of Perl/HTML Archive, 375-377
situl Insight Foundation WebABLE!, 933
situl International Cryptography, 933
situl Internet Baglady's, 508
situl ISO8859-1 (Latin 1) Table, 349, 375-377
situl Jan's Guide to HTML, 937-938
situl JavaScript, 913
situl JavaScript Index, 914
situl JavaScript Planet, 913
situl JavaScript World Web, 325, 914
situl Jerry Lemmon's Alaskan Flowers, 560
situl Learning HTML by Example, 939
situl Lexis-Nexis, 943
situl Low Vision and Deaf/Blind, 933
situl Lysator C Programming, 925
situl Malcolm's Guide to persistent Cookies, 350, 939
situl Math on the Web, 931
situl Matt's Script Archive, 375-377, 923
situl Microsoft, 479, 489
situl Names and Addresses, URIs, URLs, URNs, URCs Web, 375-377
situl Netcaster Developers Guide, 937
situl Netscape, 489
situl Object Headers, 375-377
situl Overview of HTTP, 374
situl Perl Basics, 375-377
situl Perl CGI Examples, 924
situl Perl for Win32, 375-377
situl Perl Home Page, 375-377, 924
situl Perl Tutorial, 924
situl Perl/Tk FAQ, 924
situl Persistent Cookies Documentation, 939
situl Photodisc, 508
situl Programmer's Source, 912, 924
situl Saturn Cars, 548
Situl Servlet Page, 912
situl Site Builder Network, 932
situl Special Characters, 349, 375-377
situl Start Reading, 488
situl Style Guide for Online Hypertext, 937
situl TopTen Mistakes in Web Design, 904
situl Trace R&D Center, 658
situl Travelocity, 548
situl U.S.Census Bureau, 444
situl Ultimate Tag Reference List, 937
situl Uniform Resource Locators, 375-377
situl Voodoo's Introduction to JavaScript, 914
situl W3C, 537, 816
situl W3C httpd CGI/1.1 Script Support, 375-377
situl Wagon Train Animated GIFs, 905
situl WDB - A Web Interface to SQL Databases, 375-377
situl Web Accessibility Initiative, 931
situl Web Designer Help desk, 937-938
situl Web Engineer's Toolbox, 924
situl Web Reference, 913
situl Web Reference, 932
situl Webmaster Resources, 933
situl WebTools.org, 937
situl Wilde Rose Coffee Roastary, 462
situl Wired, 498
situl World-Wide Web Virtual Library, 910
situl WWW-DBMS Gateways, 375-377
situl Xanadu, 21, 936
situl Yale C/AIM Web Style Guide, 937-938

SMTP (Simple Mail Transfer Protocol), 628
software
administratori de sit
Cold Fusion, 96
FrontPage, 895
NetObjects Fusion, 893-894
SiteMill, 893-894
tabel de comparare, 893
WebberActive, 895
editori HTML, 888-889
Amaya, 890
BBEdit, 890
Claris Home Page, 890
Composer, 890
Emacs, 893
HomeSite, 890
Hot Dog Express, 890
Hot Dog Pro, 890
HoTMetaL Pro, 890
Multi-Edit, 891-892
PageMill, 890
Vi, 893
Visual SlickEdit, 891
Webber, 35, 891
WebEdit Pro, 891
elemente de conectare, obținere, 905
I-Pad eSoft, 927
instrumente pentru maparea siturilor, 895
CLEARWeb, 896
Exploder Site Map, 897
Governor Site Map, 897
HotSauce, 897
Joe Barta's Site Map, 897
MAPA Service, 896
MapXsite, 896
SiteAnalyst, 896
SiteCommand, 897
tabel de comparare, 895
Visual Site Map, 897
WebAnalyzer, 897
WebCollection, 897
WebCutter, 897
multimedia, 906
3D Studio Max, 907
GoldWave, 909
Macromedia, 907
Movie Cleaner Pro, 909
NoteWorthy Composer, 909
Personal AVI Editor, 909
ReadAudio, 909, 906
Sound Forge, 909
tabel de comparare, 907
WebShot, 909
programe pentru grafică, 899-900
DeBabelizer, 901
Design Lab, 904
GIF Construction Set, 902
GIFTool, 903
GIFTrans, 903
GIMP, 902, 902
Illustrator, 900
Imaging Machine, 903
LView Pro, 903
Paint Shop Pro, 902
PHOTO-PAINT, 901
Photoshop, 900
tabel de comparare, 899-900
Web Wonk, 904
Web3, 902
XV Image BViewer, 903
shareware, 897
C/Net Download.com, 898
TUCOWS Central, 898
ZDNet Software Library, 898
software pentru grafică, 899-900
DeBabelizer, 901
Design Lab, 904
Fractal Design Painter, 901
Freehand Graphics Studio, 902
GIF Construction Set, 902
GIFTool, 903
GIFTrans, 903
GIMP, 902, 902
Illustrator, 900
Imaging Machine, 903
LView Pro, 903
Paint Shop Pro, 902
PHOTO-PAINT, 901
Photoshop, 900
tabel de comparare, 899-900
Web Wonk, 904
Web3, 902
XV Image BViewer, 903
sorting dynamic tables, 453-456
Sound Forge, 909
spațiere
cuvinte (spațierea după em), 821
imagini, 151-152, 514
litere, 821
text, 225
spații em, 821
spațiu
celule (tabele), 117
citate, 743
coloane, 134
design, 473
forumul DisAbilities, 558-560
imagini, 151-152, 525
spațiu alb, 49
celule (tabele), 117
definiție, 151
design, 518
alignment tricks, 520-527, 522-526, 525-526
margini, 518, 520
proporția, 526
proximitatea, 526
documente XML, 828
forumul DisAbilities, 558-560
imagini, 151-152
secțiuni de text, 743
XML, 569-570, 593
spațiul orizontal și imagini, 514, 525
spațiul vertical, imagini, 514, 525
pagini Web, 49
spațiu alb, design, 518-522, 745
text, 225
work space, design, 516-518
XML, 569-570, 593, 828
specificații de modele de conținut, 42-43
Spiderwoman Mailing List at Amazon City, 934

BIBL. CENTR. UNIV. „M. EMINESCU" IAȘI INFORMATICĂ

SQL (Structured Query Language - Limbaj structurat de interogare), 437-438
ss (format secunde), 94
SSIs (server-side include), 76
SSL (Secure Sockets Layer), 928
SSLapps, 928
SSLeay, 928
stacking layers, 305
Standard Generalized Markup Language, *vezi* SGML
standarde
ANSI (Institutul american de standarde), 36
DHTML, 379
Netscape, 379, 383-386
W3C, 379-383
ECMA Script, 392
evenimente, scopul lor, 400
hărți imagine ce țin de server, 185
HTML, 27
HTTP, 27
ISO (International Standards Operation), 36
Modelul de Obiect Component (COM), 348
Modelul de Obiect Document (DOM), 394
cerințe, 394-395
document and content manipulation, 396-397
modelul evenimentului, 397-398
structure navigation, 395-396
SGML, 37-38
standarde Unicode, 37-38, 626
ideograme, 627
text bidirecțional, 636
W3C, 27-28
foi de stil în cascadă (Cascading Style Sheets) (CSS), 230, 816
foile de stil JavaScript, 288
Inițiativa pentru Accesibilitatea Web (Wai), 381, 642-643
Meta Framework Content (MCF), 669
process for setting standards, 28
standardul CJK, 627
standardul Unicode, 37-38, 626
avantajele folosirii codului Unicode ca și set de caractere al documentului, 631
fonturi, 639
ideograme, 627
modul de scriere cursiv, 637
text bidirecțional, 636
static hyperlinks, culori, 502
stiluri (text), 238-242
stiluri condiționale în JSSS, 289
stoc de fotografii, 508
structura
documente HTML, 30, 33
antet, 31-32
cadre, 31-32
corpul, 31-32
scripturi, 31
tabele, 33
hiperlegături, 168
structure navigation (DOM), 395-396
Structured Query Language (SQL), 437-438
submulțimi interne metoda substr(), 859
submulțimi, *vezi* submulțimi externe;
subrutine
domeniu de existență, definire, 402-403
evenimentul onload, 54, 338
evenimentul onunload, 77-78
inline event handlers, 401, 409-412
Internet Explorer, 409
JavaScript, 317
obiectul area, 838
obiectul button, 838
obiectul checkbox, 839
obiectul combo, 840
obiectul document, 842
obiectul form, 843
obiectul frame, 845
obiectul image, 846
obiectul layer, 848
obiectul link, 849-850
obiectul password, 853
obiectul radio, 854
obiectul reset, 856
obiectul select, 857
obiectul submit, 859-860
obiectul text, 861
obiectul textarea, 861-862
obiectul window, 865
hărți imagine
harta imagine (NCSA), 186
htimage (CERN), 186
rutine de tratare a formularelor, 363, 368-370
subrutine de tratare a evenimentelor, 68, 400
domeniu de existență, definire, 402-403
evenimentul onload, 77-78, 338
evenimentul onunload, 77-78
inline event handlers, 401, 409-412
Internet Explorer, 409
JavaScript, 317
obiectul area, 838
obiectul button, 838
obiectul checkbox, 839
obiectul combo, 840
obiectul document, 842
obiectul form, 843
obiectul frame, 845
obiectul image, 846
obiectul layer, 848
obiectul link, 849-850
obiectul password, 853
obiectul radio, 854

obiectul reset, 856
obiectul select, 857
obiectul submit, 859-860
obiectul text, 861
obiectul textarea, 861-862
obiectul window, 865
subrutine de tratare a evenimentelor, 68, 400
subseturi externe (XML DTDs), 567
subseturi interne (XML DTDs), 567
Sun
familia server, 919-920
Java, 316
JavaScript, 316
sunet
hiperlegături la fișiere de sunet, 173-174
inserare, 220
în pagini HTML, 174
software
GoldWave, 909
NoteWorthy Composer, 909
Sound Forge, 909
suprapunere
elemente în foile de stil în cascadă, 268-269
layere, 305
text, 241-242
surse de lumină (imagini), 510
sursele citatelor, 90
SyteAnalyst, 896

Ş

şabloane
foile de stil în cascadă (CSS), 249-250
pagini Web
copiere, 61-63
creare, 56-60
şirul de date, 40
şiruri (funcţiile VBScript), 882
şiruri în VBScript funcţii de tratare a şirurilor, 870-871
ştergerea textului
data şi timpul modificărilor, 16, 94
marcarea textului ce va fi şters (editarea documentelor), 93

T

tabele, 33, 116, 132
afişare, 127-129
aliniere, 116
avantaje în designul paginilor Web, 132
cadre, 116
celule, 116
aliniere, 123
creare, 118-120
definire, 133
formatare, 116
fuzionare, 120-122
împachetarea textului, 124
lăţime, 127-128
mărime, setare, 120
padding, 117
spaţiu, 117
coloane, 116
alipire peste, 120-121
coloane de ziar, 142-144
coloane index, 136-138
creare, 133-134
grupare, 125-127
lăţimi, 134
macheta ziarului, 141-142
note, 138-139
spaţiu, 134
texte suplimentare, 138-139
conţinut
adăugare 154-156
inserare, 136, 139-140
creare, 57-58, 132
graniţe, 117
lăţimi, 116
machetare, 55-56, 528-529
navigare
folosirea marginii paginilor pentru navigare, 489-491
folosirea părţii de jos a paginilor pentru navigare, 488
folosirea părţii superioare a paginilor Web pentru navigare, 488
numere, formatare, 118
problema accesibilităţii, 648-650
reguli, 117
rânduri, 116, 133
aliniere, 118
alipire peste, 122
creating, 118
formatare, 116
grupare, 124
sintaxa, 132
soluţii pentru afişarea tabelelor în browsere care nu permit aceasta, 88
suportul browserului, 142
tabele dinamice
editare, 452
expansiune, 448-452
sortare, 453-456
viewing records, 452
tabele dinamice
editare, 452
expansiune, 448-452
sortare, 453-456
viewing records, 452
tabelul de comparare a browserelor, 736-880
caractere, 802-815
culori, 801
entităţi escape, 800
evenimente intrinseci, 799
importanţă, 734
precizie, 815
proiectare, 735-736
tabla de materii (cadre), 285-286
tastaturi
navigarea cu taste de acces, 211-212
navigarea tab, 211
problema accesibilităţii, 642-643, 656-657
târâtoare, 661
tehnici escape (codificări de la 16 biţi la 8 biţi), 626
tehnologie de sprijin, 580
tehnologii asistente, 643-644

telefoane (schema URL callto), 166
telnet, 24-25, 164
teoria bazelor de date relaţionale, 437-438
text
aliniere, 68, 242-246
alinierea la dreapta, 521
alinierea la stânga, 521
alinierea pe centru, 520-527
design, 520
foile de stil în cascadă, 241-242
anteturi, 82
atribute
atributul font-family, 238-240
atributul font-size, 239-240
atributul font-style, 240
atributul font-weight, 240
atributul letter-spacing, 821
atributul line-height, 241, 822
atributul margin-left, 241-242
atributul margin-right, 241-242
atributul margin-top, 241-242
atributul text-align, 241-242, 822
atributul text-decoration, 241, 821
atributul text-indent, 241-242, 822
atributul text-transform, 822
atributul vertical-align, 822
atributul word-spacing, 821
boldfacetext, 115
bounding boxes, 822
celule (tabele), împachetarea textului în jur, 124
citate, 90
citate, formatare, 94-96
cod, formatare, 91-93
coloane (coloane stil ziar), 142-144
culoare, 502
culori, 228
definiţii, 90
foi de stil, *vezi* foile de stil
fonturi
applying, 534
mărimi, 535-537, 537
netezire, 537
formatare, 222
barat, 224
boldface, 223
exponent, 225-226
fonturi, 226-228
indice, 225-226
italice, 223
marcarea textului inserat, 224
mărime, 224
subliniat, 223-224
text monospaţiat, 91-93, 225
hyperlinks for navigation, 484
imagini, alining images with text, 523
indentări, 241-242
interliniere, 241, 537
italice, 115
întreruperi de rând, creare, 85-87
list of explanatory text, 103
machetare, 49-50, 59-60
marcaje pentru editare, 93-94
margini, 241-242
modificarea spaţiului dintre litere, 537
modul de scriere cursiv, 637-638
paragrafe
aliniere, 85
creare, 84-85
plutirea în jurul imaginilor, 85-86
problema accesibilităţii, 645
acronime, 90
adrese, formatare, 94
culori, 646-647
formatare, 645
liste, 647-648
marcare, 645
problemele browserului, 83
scriere cu majuscule, 822
selectare, 408
separatori (linii orizontale), 96
spaţii em, 821
suprapunere, 241-242
tabele
adăugare, 136
inserare, 136, 139-140
text bidirecţional, 636
anulare, 637
atributul dir, 637
RFC2070, 636-637
Unicode, 636
text preformatat, 87-88
text vertical, 537
valori
valoarea LTR (left-to-right), 68
valoarea RTL (right-to-left, 68
writing tips, 471
text barat, 224
text BIDI, 636
anulând, 637
atributul dir, 637
text bidirecţional, *vezi* text BIDI
RFC 2070, 636-637
Unicode, 636
text evidenţiat, 89, 223
text exponent, 225-226
text indice, 225-226
text monospaţiat, 91-93, 225, 239, 532-533, 818
text preformatat, 87-88
text subliniat, 223-224
text teleimprimator, 225
text vertical, 537
aliniere, 241-242
texte suplimentare, creare, 138-139
tipografie
design, 532-537,

forumul DisAbilities, 560-562
vezi și fonturi
tipul caracterelor și machetarea, 537
tipul ENTITIES (XML), 569-570
tipul ENTITY (XML), 569-570
tipul ID (XML), 569
tipul IDREF (XML), 569-570
tipul NMTOKEN (XML), 569-570
tipul NMTOKENS (XML), 569-570
tipurile enumerare (atributele XML), 830
tipurile enumerate (atributele XML), 569-570
tipurile media (ieșire), 74
tipurile string (atributele XML), 569, 830
tipurile token (tipurile simbolice) (atribute XML), 569-570, 830
titluri, 57-58, 71-72
titluri multiple, 72
writing tips, 72
top frames, 281
trafic, vedeți vizitatori
translating encoded forms, 195
transmiterea formularelor (completarea formularelor), 196, 358
transparență
imagini, 513
layere, 302
tratarea erorilor (funcții VBScript), 874-875
trecerea în revistă a capitolelor în cadre, 285-286
triplet hexazecimal (culori), 77-78
triunghiuri, design, 531
TUCOWS Central, 898
Tucson Weekly, 492
TypeStyle, 238
TZD (time zone designator), 94

U

U.S.Departament of Defense, 36
UDIs, 156
umbre (imagini), 510
umbre gradate (imagini), 510
Unified Web Site Accessibility Guidelines, 645
Universal Coordinated Time (UTC), 94
Universal Document Identifiers, *vezi* UDIs
UNIX
CGI (Common Gateway Interface), 353
whirlGIF, 220
unloading Web pages, 404
updating search engines, 662
uploading files for forms, 201-202
URIs (Identificatori universali de resurse), 156
URL, 156
caractere, 165
DTDs, 40
fragment URLs, 160-161
gazde, 156
legături XML, 576, 834
origini, 22
scheme, 161
callto, 166
file, 164
FTP, 164
HTTP, 161-162
mailto, 162-163
news, 163
NNTP, 163
RealAudio, 173-174
telnet, 164
TV, 165
URL-uri de bază, 158-160
URL-uri relative, 157-158
URL-uri tip mailto, 195, 737-738
URL-uri parțiale, 157-158
URL-uri relative, 157-158
URL-urile mailto (adrese URL de tip mailto), 161-163, 195, 737-738
URL-urile news, 161, 163
URLs (Locatori uniformi de Resurse), 156
URNs (nume universale de resurse), 156
user input
Java Script, 319, 322
VBScript, 346-347
USER-AGENT (antete HTTP), 355
UTC (Universal Coordinated Time), 94
utilități și animația GIF, 219
utilizatori
evenimente
evenimentul Microsoft Internet Explorer, 409
evenimentul onblur, 407
evenimentul onclick, 405
evenimentul ondblclick, 405
evenimentul onfocus, 407
evenimentul onkeydown, 407
evenimentul onkeypress, 407
evenimentul onkeyup, 407
evenimentul onload, 404
evenimentul onmousedown, 405
evenimentul onmousemove, 406
evenimentul onmouseout, 406
evenimentul onmouseover, 406
evenimentul onmouseup, 405
evenimentul onreset, 408
evenimentul onselect, 408
evenimentul onsubmit, 408

evenimentul onunload, 404
interfețe, 460
analyzing purpose of, 462-467
anteturi, 470
componente comune, 466-470, 473
contact information, 471
identificarea sitului, 471
implementarea proiectului, 466
navigare, 471
principiile proiectării, 460-463, 472, 474-475
scop, 460
spațiu alb, 473
tips for writing text, 471
numărătoare de acces, 361-363
utilizatori surzi și elemente de accesibilitate, 643-644

V

valoarea absmiddle (atributul align), 522
valarea checkbox (atributul type), 198-199
validare
documente XML, 564, 590-591
în documente XML față de documente HTML, 583-585
validatori (HTML), 915-916
CAST Bobby 2.0, 917
CSE 3310, 916
tabel de comparare, 916
Weblint, 917
WebTechs, 916
validatorul HTML CSE 3310, 916
valoarea baseline
atributul align, 522
vertical-align attribute, 821
valoarea blink (atribut de decorare a textului), 821
valoarea bottom (atributul vertical-align), 821
valoarea button (atributul type), 197
valoarea hidden (atributul type), 202
valoarea image (atributul type), 198
valoarea italic (atributul font-style), 819
valoarea line-through (atributul text-decoration), 821
valoarea LTR (left-to-right), 68
valoarea middle (atributul vertical-align), 821
valoarea none (atributul text-decoration), 821
valoarea normal (atributul font-style), 819
valoarea oblique (atributul font-style), 819
valoarea overline (atributul text-decoration), 821
valoarea password (atributul type), 203-204
valoarea radio (atributul type), 200-201
valoarea reset (atributul type), 198
valoarea RTL (right-to-left), 68
valoarea sub (atributul vertical-align), 821
valoarea submit (atributul type), 198
valoarea super (atributul vertical-align), 821
valoarea text (atributul type), 202-203
valoarea text-bottom (atributul vertical-align), 821
valoarea text-top (atributul vertical-align), 821
valoarea top (atributul align), 522
valoarea top (atributul vertical-align), 821
valori
text
RTL (right-to-left), 68
valoarea LTR (left-to-right), 68
valori hexazecimale (culori), 246, 250, 505-507
atributele element, 45-46
valorile first-letter, 816
vaporware (Xanadu), 21
variabila de mediu AUTH TYPE(CGI), 356
variabila de mediu CONTENT LENGTH (CGI), 356
variabila de mediu CONTENT TYPE (CGI), 356
variabila de mediu GATEWAY INTERFACE (CGI), 356
variabila de mediu PATH INFO (CGI), 356
variabila de mediu REMOTE ADDR (CGI), 357
variabila de mediu REMOTE HOST (CGI), 357
variabila de mediu REMOTE IDENT (CGI), 357
variabila de mediu REMOTE USER (CGI), 357
variabila de mediu REQUEST METHOD (CGI), 357
variabila de mediu SCRIPT NAME (CGI), 357
variabila keywords (atributul http-equiv), 76
variabila refresh (atributul http-equiv), 76
variabila reply-to (atributul http-equiv), 76
variabile
formatarea textului, 92
variabila de mediu (CGI), 357
variabile de mediu (CGI), 356
variabile de mediu QUERY STRING (CGI), 357

variabile de mediu SERVER NAME (CGI), 357
variabile de mediu SERVER PORT (CGI), 357
variabile de mediu SERVER PROTOCOL (CGI), 357
variabile de mediu SERVER SOFTWARE (CGI), 357
variante (VBScript), 333
VBScript, 317, 332-333
ActiveX, 347-348
browsere
ascunderea scripturilor de browserele incompatibile, 335-336
rularea scripturilor, 338-339
suport, 332-335
cod, inserare, 337-338
constante, 871
declarații
Call, 877-878
Dim, 870-871, 875
Do...Loop, 877-878
End, 875
Erase, 870
Exit, 875
For... Next, 877-878
Function, 875
If... Then ... Ise, 877-878
Option Explicit, 877-878
Preserve, 871
Randomize, 870
Rem, 871
Select Case, 881
Set, 871
Sub, 875
Ubound, 871
While...Wend, 881
distincție între majuscule și litere mici, 427
documente HTML, înglobare, 335-338
etichete HTML, 418
formulare, validarea datelor, 340-343
funcții
Abs, 872
Asc, 872
Atn, 870
CBool, 872
CByte, 872
CDate, 872
CDbl, 872
Chr, 872
CInt, 872
CLng, 872
Cos, 870
CSng, 872
CStr, 872
Date, 873
DateSerial, 873-874
DateValue, 873-874
Day, 875
Exp, 870
Fix, 873
funcția Split, 334
Hex, 873
Hour, 875
InputBox, 876
InStr, 881
Int, 873
IsArray, 870, 879-880
IsDate, 879-880
IsEmpty, 879-880
IsNull, 879-880
IsNumeric, 879-880
LBound, 870
LCase, 881
Left, 881
Len, 881
Log, 870
LTrim, 881
Mid, 881
Minute, 875
Month, 875
MsgBox, 876
Now, 875
Oct, 873
Right, 881
Rnd, 870
RTrim, 881
Second, 875
Sgn, 873
Sin, 870
Space, 879-880
Sqr, 870
StrComp, 879-880
String, 879-880
Tan, 870
Time, 875
TimeSerial, 873-874
TimeValue, 873-874
Trim, 879-880
UCase, 879-880
VarType, 879-880
Weekday, 875
Year, 875
future, 392
listă poștală, 934
litere, 871
mesajele din bara de stare, 343-345
Modelul de Obiect Document (DOM), 392
Modelul de Obiect pentru Scriptare Internet Explorer, 338-340
operatori, 876-877
plug-in ScriptActive (Netcape), 334
resurse, 349
resurse online, 915
cărți, 350
grupuri de discuții, 350
liste poștale, 350
resurse online, 348-350
securitate, 334
strengths, 340
tratarea erorilor, 874-875
variante, 333
viruși, 334
Visual Basic, 333-334
Veronica, 24-25
vertical bar connector, 43
Vi, 893
Virtual Library of WWW Development, 933
viruși (VBScript), 334
vision-impaired users and accessibility issues, 642-643, 646
visited hyperlinks, colors, 502
Visual Basic
CGI (Comon Gateway Interface), 353
resurse online, 925-926
VBScript, 317, 333-334
Visual Basic Scripting, *vezi* **VBScript**

Visual JavaScript 1.0, 327, 914
Visual Site Map, 897
Visual SlickEdit, 891
viteza de încărcare, creștere, 59, 61, 184
vizitatori
formulare de răspuns, 469
motoare de căutare, rol, 660
VRML and JavaScript, 322-323

W

W3C (Consorțiul World Wide Web), 930-932
browserul Jigsaw Web server, 216
DHTML
Modelul de Obiect Document, (DOM), 379-383
standarde, 379-383
grafică pe pagina Web, 931
HTML 4.0 Specification W3C Working Draft, 671
HTML Working Group, 14
încapsularea fonturilor, 639-640
Modelul de Obiect Document, 381, 394-399, 930
pagina CSS, 930
pagina HTML 3.2, 930
pagina Inițiativa pentru Accesibilitatea Web (WAI), 381, 642-643
pagina Web Math on the Web, 931
pagina XML, 932
standarde, 28
DHTML, 379-383
foi de stil în cascadă (Cascading Style Sheets), (CSS), 230, 816
foi de stil JavaScript, 288
Meta Framework Content (MCF), 669
Modelul de Obiect Document (DOM), 394-399
WAI (Inițiativa pentru Accesibilitatea Web), 381
WAIS (Wide Area Information Server), 24-25
WDVL (Web Developer's Virtual Library), 933
WDVL (Web Developer's Virtual Library), 933
Web Embedding Font Tool (Microsoft), 535
Web Wonk, 904
Web, definire, 23
WebAnalyzer, 897
Webber32, 891
WebberActive 1.1, 895
WebCollection, 898
WebCutter, 897
WebEdit Pro, 891
Weblint, 917
WebShot, 909
WebStar 2.1, 918-919
WebTechs validation service, 916
Weinman, Lynda, 498
Wheildon, Colin, 537
whirlGIF, 220
Wide Area Information Server, *vezi* **WAIS**
Will-Harris, Daniel, 238
William, Robin, 537
Windows Alchemy Mindworks GIF Construction Set, 219
WinHelp, 24
Wood, Lauren, 382
word processors an form decoding, 196
World Wide Web, 22
WWW (World Wide Web)
AltaVista, 941
căutare, 940-941
Excite, 941
Infoseek, 941
Lexis-Nexis, 943
Lycos, 943
Magellan, 943
mecanisme de căutare, 941
Reuters, 943
tabel de comparare, 941
Yahoo!, 943
WWW Security FAQ, 927

X

XLL (limbajul de legare XML), 572
XML, 564
aplicații
Cartea de răspunsuri, 580
Channel Definition Format (CDF), 580
limbajul de marcare chimic (CML), 580
limbajul de marcare cu dispozitiv (HDML), 580
limbajul de marcare matematic (MathML), 580
Open Software Description (OSD), 580
Open Tag, 580
atribute
atributele tip enumerat, 569-570
atributele tip șir, 569
atributele tip token, 569-570
atributul ACTUATE, 575, 833
atributul CONTENT-ROLE, 574
atributul HREF, 575, 833
atributul href, 573
atributul INLINE, 574-575, 833
atributul ROLE, 574-575, 833
atributul SHOW, 575, 833-834
atributul TITLE, 574-575, 833
atributul XML-ATTRIBUTES, 832-833

atributul XML-LINK, 573, 578
atributul XML-SPACE, 828
declarații, 829
prestabilite, 830
tipurile enumerare, 830
tipurile simbolice, 830
tipurile string, 830
XML-LINK, 832-833
comentarii, 566
comparare cu HTML, 582
avantaje, 582
foi de stil, 587-588
legare, 585-587
validare, 583-585
declararea solicitată a marcării (RMD), 584
documente, 572, 595, 827
caractere, 827
comentarii, 827
declarare, 566-567
formatul corespunzător, (buna-formare), 565, 590-591, 827
instrucțiuni de procesare, 828
prologuri, 828
referințe de caracter, 832
referințele entitate, 832
Required Markup Declaration, 828
secțiunile CDATA, 828
spațiul alb, 828
structura fizică, 827
structura logică, 827
validarea, 564, 590-591
DTDs, 567
creare, 591-596
declararea elementelor, 567-568
declarațiile atributelor, 569-570
declarațiile entităților, 570-571
secțiuni condiționale, 572, 830-831
secțiunile CDATA, 572
sintaxa, 828
subseturi externe, 567
subseturi interne, 567
XSL, 621
elemente
declarații, 829
structuri fizice, 830
structuri logice, 830
entități, 565
declarații, 831-832
entitatea document, 831-832
entități binare, 570, 594, 597-598, 831
entități externe, 571
entități interne, 570
entități parametru, 570, 594
entități predefinite, 572, 832-833
entități text, 570, 594-597, 831
referințe, 830
reguli de procesare XML, 832-833
structuri fizice, 830
structuri logice, 830
etichete
etichete de început, 566, 829
etichete de sfârșit (etichete de încheiere), 566, 829
structuri fizice, 831
structuri logice, 831
foile de stil în cascadă (CSS), 578
legături
adresare, 575, 834
asocierea cu informația, 574-575, 833
atributul href, 573
candidați, 576-577
comportare, 575, 834
domeniile pointerilor extinși, 577
grupuri de legături extinse, 835
identificator de fragment, 577
legături extinse, 574, 598-599, 833
legături inline, 574, 834
legături simple, 573, 833
limbajul de legare XML, 572
pointeri extinși (indicatori extinși), 576, 834-835
recunoașterea legăturilor, 832-833
remaparea atributului, 833-834
resurse, 572
observații, 832-833
referințele de caracter, 832
reședință Web, 932
spațiu alb, 593
viitor, 580-581
XSL, 577-578, 600
anularea foilor de stil, 616-617
DSSL, 579, 600-601
ECMAScript, 579, 617-619
foi de stil cu structură fixă, 620-621
foi de stil în cascadă (CSS), 600
functionalitate, 579, 601
importul foilor de stil, 616
obiectele de curs, 601-605
procesare, 579-580
reguli de construcție, 579, 605-613
reguli de stil, 606, 614-615
XML DTDs, 621
XSL, 577-578, 600
atribute, 608-609
calificative, 609-610
priorități, 610
anularea foilor de stil XSL, 616-617
DSSL, 579, 600-601
ECMAScript, 579, 617-619
exemple, 617-619
folosire, 617-618
specificații, 617-618

foi de stil cu structură fixă, 620-621
foi de stil în cascadă (CSS), 600
funcționalitate, 579, 601
importul foilor de stil XSL, 616
obiectele de curs, 601-605
caracteristici, 604
macroinstrucțiuni, 613-614
obiectele de curs DSSL, 602-604
obiectele de curs HTML/CSS, 604
procesare, 579-580
reguli de construcție, 579, 605-613
acțiuni, 610-612
modele, 607-608
reguli de stil, 606, 614-615
moduri, 615-616
numele stilurilor, 615
XML DTDs, 621
XV Image Viewer, 903

Y-Z

Yahoo!, 350, 375-377, 661, 943
YYYY (format an), 94
ZDNet's Software Library, 898, 909
ziare (online)
Albuquerque Alibi, 492
Tucson Weekly, 492

BIBL. CENT. UNIV. „M. EMINESCU" IAȘI INFORMATICĂ